commodore Magazine

AÑO 2 · Núm. 22
Diciembre 1985 · 300 pts.

REVISTA INDEPENDIENTE PARA USUARIOS



ORDENADOR PERSONAL

Canon

V-20

SISTEMA MSX

Y ADEMAS
PUEDE VIAJAR
A MEJICO
CON LA SELECCION
PREGUNTE
A SU PROVEEDOR

DISFRUTELO EN FAMILIA

El ordenador personal para toda la familia. Con 80 K para estar a la altura de todos los gustos y necesidades. Jugar, archivar, aprender, programar: y con capacidad para crecer con la aplicación de periféricos.

CARACTERISTICAS MAS IMPORTANTES DEL V-20

- Sistema standard MSX que hace compatibles hardware y software de todos los productos que tienen este sistema en el mercado.
- Pueden acoplar los siguientes periféricos de CANON:
  - — Impresoras.
  - — Floppy de 720 K, que incluye diskette MSX-DOS para aprovechar toda la capacidad del ordenador y además incluimos un segundo diskette con tres programas de aplicaciones profesionales.
  - — Mouse para hacer todo tipo de gráficos a color.
  - — Joy sticks.
  - — Caja de 5 diskettes vírgenes.
  - — Variedad de programas en cinta con juegos.
- Y además dos manuales en castellano: guía del usuario y completo manual de BASIC.

commodore Magazine

Año 2 Núm.22

Director:
Roberto Menéndez
Coordinadora editorial:
Sonia Ortega
Redacción:
Mario Cavestany
Ignacio Martín
Teresa Aranda
Fernando García
Diseño:
Ricardo Segura
Benito Gil

Editada por
Publinformática
Presidente:
Fernando Bolín

Gerente de Circulación y Ventas:
Luis Carrero
Producción:
Miguel Onieva
Director de Marketing:
Antonio González
Servicio al cliente:
Julia González - Tel.: 733 79 69
Administración:
Miguel Atance
Jefe de Publicidad:
María José Martín
Dirección y Redacción
C/Bravo Murillo, 377 - 5.º A
Tel.: 733 74 13
28020 - MADRID
Publicidad y Administración:
C/ Bravo Murillo, 377 - 3.º E
Tel. 733 96 62/96
Publicidad en Madrid:
Fernando Hernando
Publicidad en Barcelona:
María del Carmen Ríos
Pelayo, 12.
Tel.: (93) 301 47 00 ext. 27 y 28
08001 - BARCELONA
Depósito Legal: M-6622-1984
Distribuye: S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas. Madrid
Distribuidor en Venezuela:
SIPAM, S.A.
Avda. República Dominicana
Edif. Feltrec
Boleita Sur Caracas (Venezuela)
Fotocomposición: Consulgraf
C/Nicolás Morales, 34. Madrid
Fotomecánica: Karmat
C/Pantoja, 10. Madrid
Imprime: G.Velasco S.A.
Solicitado control O.J.D.
Esta publicación es miembro
de la Asociación de Revistas
de la Información ARI, asociada
a la Federación Internacional
de Prensa Periódica FIPP.

Rogamos dirijan toda la correspondencia relacionada con suscripciones a Commodore Magazine. Edisa. Tel.: 415 97 12. López de Hoyos, 141-5 28002 Madrid. Para todos los pagos reseñar solamente Commodore Magazine.
Para la compra de ejemplares atrasados dirigirse a la propia editorial Commodore Magazine. Bravo Murillo, 377 5° A. Tel.: 733 74 13. 28020 Madrid.

Editorial

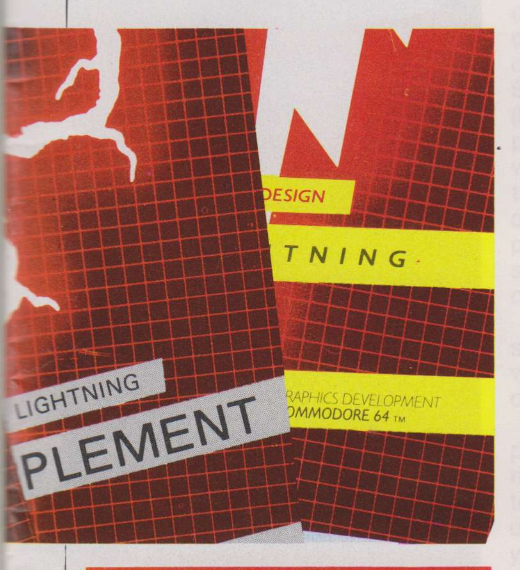
El pasado mes de Noviembre, entre los días 15 y 22, tuvimos ocasión de asistir al SIMO, sin duda una de las más importante ferias del sector informático de nuestro país. Commodore, de la mano de su distribuidor Microelectrónica y Control, presentaba al público el nuevo modelo C-128, junto con las compatibles IBM PC10 y PC20. El 128, que se presentaba al lado de varios C-64, es una máquina que lleva tiempo anunciada y que está despertando unas enormes expectativas tanto entre los usuarios de Commodore como entre los que están pensando en la adquisición de un nuevo microordenador. Sus característica hacen de ella un modelo muy atractivo, sobre todo el aspecto de la compatibilidad, tanto de hardware como de software, con el C-64 y la estupenda versión de BASIC incluída en la ROM de esta máquina. Sin embargo, aunque presentada en esta feria y ya comentada en nuestras páginas el pasado mes de noviembre, no se espera su comercialización hasta el mes de enero, lo que supone a nuestro modo de ver un retraso importante. Su precio, según las noticias de que disponemos, andará entorno a las ochenta o noventa mil peseta, situándola a un buen nivel de precio con respecto a sus competidores inmediatos. Por lo que respecta al contenido de este mes incluímos un comentario acerca de una serie de paquetes software orientados al desarrollo de programas y denominados programas LIGHTNING. Su interés es enorme y sus capacidades y posibilidades, impresionantes, lo que puede hacer de ellos un buen objetivo para estas próximas navidades. Los tres programas que comentamos incluyen una versión BASIC, un version FORTH y un macroensamblador. Otro aspecto que queremos comentar y que puede dar una idea de la potencia del mercado de los juegos es la gran cantidad y calidad de los programas de juego que están llegando hasta nosotros. El Commodore 64 sigue siendo la mejor máquina de juegos del mundo. Además, aunque por ahora el porcentaje de juegos de calidad hechos en España es prácticamente nulo, tenemos noticias de que alguna firma de software va a comercializar programas nacionales con unos niveles de calidad comparables a los de los programas extranjeros. □

Esta revista no mantiene relación de dependencia de ningún tipo con respecto de los fabricantes de ordenadores Commodore Business Machines ni de sus representantes.

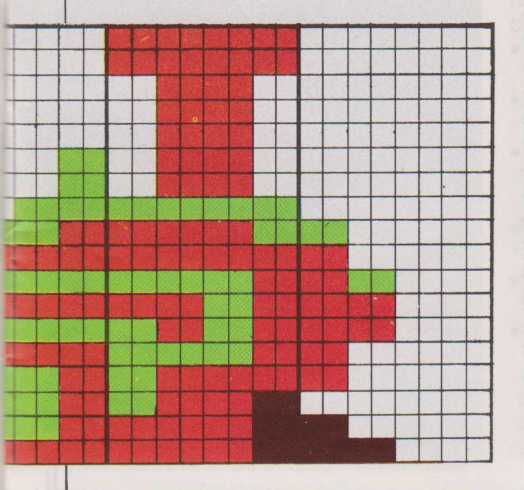
Codigos de control para el VIC-20 y el C-64

COMO SE VE	COMO SE TECLEA	EFECTO CONSEGUIDO
COLORES DEL VIC-20 Y DEL 64		
	CTRL+1	NEGRO
	CTRL+2	BLANCO
	CTRL+3	ROJO
	CTRL+4	CIAN
	CTRL+5	PURPURA
	CTRL+6	VERDE
	CTRL+7	AZUL
	CTRL+8	AMARILLO
COLORES DEL 64 SOLAMENTE		
	CBM+1	NARANJA
	CBM+2	MARRON
	CBM+3	ROSA
	CBM+4	GRIS OSCURO
	CBM+5	GRIS MEDIO
	CBM+6	VERDE CLARO
	CBM+7	AZUL CLARO
	CBM+8	GRIS CLARO
CODIGOS DE CURSOR Y CONTROL		
	HOME	CURSOR A CASA
	SHIFT+HOME	LIMPIA PANTALLA
	CRSR	CURSOR DERECHA
	SHIFT+CRSR	CURSOR IZQUIERDA
	CRSR	CURSOR ABAJO
	SHIFT+CRSR	CURSOR ARRIBA
	CTRL+9	CARACTER INVERSO
	CTRL+0	CARACTER NORMAL
	DEL	BORRAR
	SHIFT+DEL	INSERTAR
TECLAS DE FUNCION		
	F1	
	F2=SHIFT+F1	
	F3	
	F4=SHIFT+F3	
	F5	
	F6=SHIFT+F5	
	F7	
	F8=SHIFT+F7	

SUMARIO



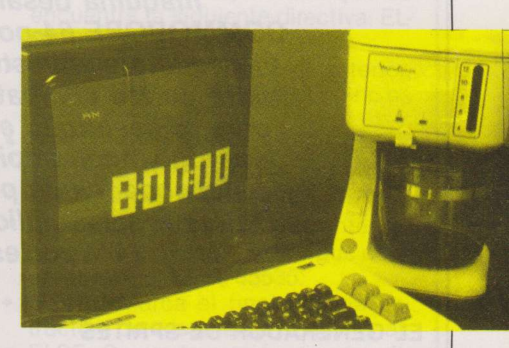
DESIGN
TNING
LIGHTNING
PLEMENT
RAPHICS DEVELOPMENT
COMMODORE 64 TM





commodore





8:00:00

«Machine lightning»

Vamos a comentar aquí un potente programa para aplicaciones en código máquina desarrollado para el COMMODORE 64 por OASIS SOFTWARE. El paquete suministrado se compone de cuatro partes compatibles, y está pensado para el desarrollo eficaz de juegos. Sin embargo, nada impide usarlo como una estupenda ayuda para el desarrollo de nuestras propias aplicaciones. Estas cuatro partes son:

EL GENERADOR DE SPRITES

Con el que podremos diseñar y editar los gráficos que vamos a usar en nuestro juego.

BASIC LIGHTNING

Una extensión al intérprete de BASIC suministrado en el COMMODORE 64. Ofrece, además de las ventajas de la programación estructurada (procedures, IF...THEN...ELSE, lazos DO WHILE, CASE, etc.) una serie de comandos correspondientes a una librería de rutinas gráficas de **MACHINE LIGHTNING**. Eso nos permite llamarlas directamente desde el BASIC.

64-MAC/MON

Es un Macroensamblador/monitor completo para escribir nuestras aplicaciones en lenguaje máquina. Es la parte que vamos a comentar primordialmente en este artículo.

LAS RUTINAS GRAFICAS

Son 10 K de código máquina con rutinas para manejar sprites en la pantalla. Proveen funciones tales como poner y quitar sprites, scroll, variación de tamaño, detección de colisiones, así como la definición de nuestros propios sprites software.

64-MAC/MON

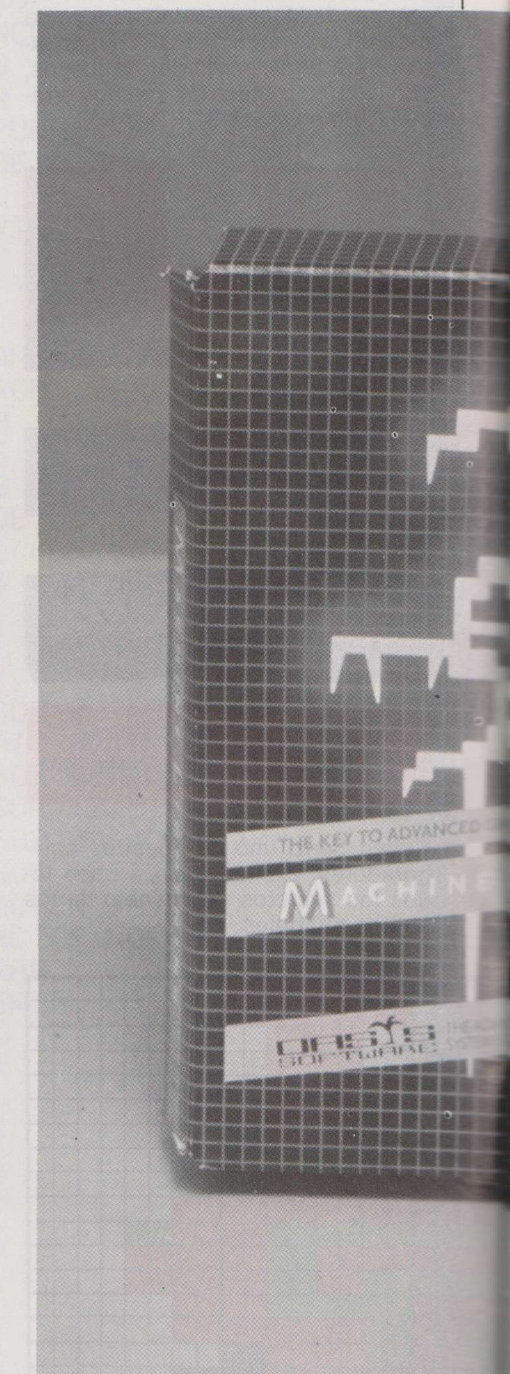
Con 64-MAC/MON seremos capaces de desarrollar y depurar complejos programas en lenguaje máquina para nuestro COMMODORE 64.

Esta herramienta nos permite el uso de aproximadamente 70 comandos para escribir y depurar programas en ensamblador. Incluye un editor de línea para la creación del programa fuente, un macroensamblador completo de dos pasos, un desensamblador, un monitor código máquina y un depurador.

EL MACROENSAMBLADOR

Puede operar en modo residente o ensamblando código fuente desde el disco. El ensamblaje es rápido (aproximadamente a una velocidad de 20.000 líneas por minuto). Cuando ensamblamos desde memoria, podremos mantener programas de hasta aproximadamente 2.500 líneas.

En modo disco de pueden ensamblar programas leyéndolos directamente desde el diskette. Se ofrece además la posibilidad de ensamblar varios ficheros concatenándolos uno detrás de otro. El tamaño del código fuente que puede ensamblarse, en este caso, está limitado únicamente por la cantidad de almacenamiento disponible en el disco; pueden así alcanzarse aproximadamente hasta 8.500 líneas de código en el caso de la uni-





dad de disco típica 1541. Se incluyen también facilidades tales como ensamblaje condicional, referencias cruzadas y facilidades para la paginación en la impresora.

Antes de pasar a analizar brevemente las opciones y directivas que ofrece este macroensamblador, explicaremos brevemente qué es un macroensamblador.

Un macroensamblador no es más que un ensamblador de código máquina tradicional que permite el uso de macros. ¿Qué son estas macros? Son el equivalente en ensamblador (bajo nivel) a lo que las subrutinas, procedures y functions son en alto nivel. Es decir, son un conjunto de instrucciones escritas en ensamblador y que pueden ser llamadas en cualquier parte del programa principal, con una serie de parámetros para efectuar una cierta función.

Vamos a examinar ahora las posibilidades del macroensamblador, viendo las directivas y opciones que ofrece.

Veamos en primer lugar las principales directivas. Estas son instrucciones que no forman parte propiamente de nuestro programa, sino que le dicen al ensamblador de qué manera y con qué opciones tiene que efectuar el ensamblaje. Una sentencia directiva consta de los siguientes elementos:

<etiqueta> <directiva>
<operando> <comentario>

La etiqueta y el comentario son opcionales y en algunos casos no se requiere el campo de operando.

A continuación damos una somera descripción de las directivas principales:

- BYTE: Se usa para definir constantes que ocupan un byte.
- DBYTE: Se definen constantes que ocupan dos bytes.
- WORD: Lo mismo que DBYTE pero los dos bytes se toman en orden inverso.
- PAD: Rellena un área de memoria con NOP.
- END: Fin del programa a ensamblar.
- BLOCK: Se usa para reservar espacio en memoria.

=: Iguala una etiqueta a una expresión en tiempo de ensamblaje.

*: Símbolo reservado para el contador de posiciones.

- ORG: Sitúa el origen del programa.
- DEFMAC: Comienzo de la definición de una macro.
- ENDMAC: Fin de la definición de una macro.
- IFEQ: Si la expresión que sigue es cero, se sigue el ensamblaje normalmente, en caso contrario se suprime el ensamblje hasta que se encuentra la siguiente directiva. ELSE o .IFEND.
- IFNEQ: Lo mismo que el anterior, pero si la expresión que sigue no es cero.
- IFPOS: Lo mismo que el anterior, pero si la expresión que sigue está en el rango de 0 a 32767.
- IFNEG: Lo mismo que la anterior, pero si la expresión que sigue está en el rango 32768-65535.
- IFEND: Marca el final de una estructura de ensamblaje condicional.
- ELSE: Segunda opción en una directiva .IF.
- PRINT: Para imprimir cadenas durante el ensamblaje.
- LIST: Activa la generación del listado de ensamblaje.
- NOLIST: Desactiva la generación del listado de ensamblaje.
- PAGE: Comienza una nueva página en la impresora.
- SKIP: Salta un número de líneas en la impresora.
- TITLE: Escribe un título en cada página de salida de la impresora.
- WIDTH: Número de caracteres por línea en la impresora.
- HEIGHT: Número de líneas por página en la impresora.
- FILE: Se usa para ensamblar un número seguido de ficheros en modo disco.

Otras entradas al ensamblador serán las sentencias con las instrucciones del 6510. Estas tienen el formato:

<etiqueta>
<código de operación>
<operando> <comentario>

USO DEL EDITOR DE LINEA

Usaremos el editor de línea para crear nuestros programas fuente que luego serán ensamblados.

MACHINE LIGHTNING nos permite definir las teclas de función del COMMODORE 64 de la siguiente forma:

Fn= <cadena>



De esta menera nos ahorramos escribir cadenas que vamos a usar muchas veces.

La forma de introducir las sentencias con el editor es muy parecida a la del BASIC: un número de sentencia seguido de la propia sentencia. Sin embargo aquí los números de sentencia no tienen que ver con el orden de ejecución, como en el BASIC.

A continuación ofrecemos un resumen de los principales comandos del editor, con su abreviatura entre paréntesis. EDITOR (ED.): Pone el ensamblador en modo 'EDITOR'. Se interrumpe el chequeo de las líneas que vamos entrando. Entrar o salir de este modo destruirá cualquier texto que hubiera en memoria.

RESIDENT (RES.): Pone el ensamblador en modo residente, en el cual se puede ensamblar texto directamente desde memoria.

DISK (DI.): Pone el ensamblador en modo 'DISK', lo que permite ensamblar programas desde el disco.

LIST (LI.): Lista líneas del texto. Le siguen las especificaciones de la línea inicial y final. Lista también los números de línea.

PRINT (P.): Lo mismo que el anterior pero no lista los números de línea.

DELETE (D.): Borra líneas del texto fuente. Pregunta si estamos seguros.

RENUMER (R.): Renumera líneas en el texto fuente.

MEM (M.): Devuelve la cantidad de memoria disponible.

NEW (N.): Borra el programa en memoria.

AUTO (AU.): Va numerando automáticamente las líneas.

MANUAL (MA.): Desactiva el modo AUTO.

MOVE (MO.): Mueve un bloque de líneas de una parte a otra del texto.

COPY (CO.): Igual que MOVE, pero las líneas no se borran de su posición inicial.

FIND (FI.): Encuentra la posición de una secuencia de caracteres en el texto.

CHANGE (CH.): Cambia todas las ocurrencias de una serie de caracteres en el texto por otra.

LOAD (LO.): Carga ficheros fuente a memoria.

SAVE (SA.): Envía código fuente a cinta o disco. El texto se comprime.

FSAVE (F.): Como SAVE, pero el texto se envía sin compresión, como si fuera a ser impreso.

MOLAD (ML.): Carga ficheros código máquina.

MSAVE (MS.): Almacena código máquina en cinta o disco.

OLOAD (OL.): y OSAVE (OS.): Lo mismo que los anteriores pero con código objeto.

MACHINE LIGHTNING también posee comandos para manejar cómodamente la impresora y los ficheros en el disco, de forma fácil y rápida (centrado y formateado de texto en la impresora, formateado, borrado, copia e impresión de ficheros en disco, etc.).

COMANDOS DE USO DEL ENSAMBLADOR

Estos comandos permiten el ensamblado de programas desde memoria o desde el disco. Entre ellos, merecen citarse los siguientes:

ASM (A.): Ensambla un fichero fuente. Puede ser seguido por letras, que marcan las siguientes opciones:

- L: Se genera un listado completo del ensamblaje.
- M: Se ensambla directamente a memoria.
- O: Ensambla código objeto a cinta o disco.
- C: Se genera un listado de concordancia (etiquetas en orden alfabético y en orden numérico).

OFFSET (O.): Offset a añadir al contador de posiciones cuando se guarda código fuente o código objeto.

RUN: Ensambla y ejecuta un programa en ensamblador.

EL MONITOR CODIGO MAQUINA

Esta parte de MACHINE LIGHTNING nos permite inspeccionar y modificar la memoria directamente, así como los registros ayudándonos de esta forma a observar los resultados y a la depuración de nuestros programas. A continuación exponemos los principales comandos:

DECIMAL (DE.): A partir de ahora toda salida numérica es en base 10.

HEX (H.): A partir de ahora, toda la salida es en base 16.

CALC (?): Evalúa una expresión e imprime el resultado en ASCII, binario, octal, decimal y hexadecimal.

MLIST: Muestra el contenido de la memoria en modo numérico y en ASCII.

MDUMP: Similar al anterior, pero no se imprime el ASCII.

MFIND (MF.): Muestra las direcciones de todas las ocurrencias de una serie de bytes entre las dos direcciones que se le indiquen.

COMPARE (CO.): Compara un bloque de memoria con otro, mostrando las diferencias.

MFILL (FILL): Rellena un bloque de memoria con una secuencia de uno o más bytes.

MMOVE (MM.): Mueve un bloque de memoria de un lugar a otro.

RELOC (REL.): Similar a MMOVE, pero se cambian las direcciones que hagan falta para que el programa corra en su nueva dirección.

MCHANGE (MCH.): Encuentra todas las apariciones de una serie de bytes y los reemplaza con otra serie, entre las direcciones de memoria que se le indiquen.

EL DESENSAMBLADOR

Esta parte de MACHINE LIGHTNING consiste en un completo de-

NUMERO 22

sensamblador que ofrece además la opción del desensamblado de los símbolos (etiquetas). Disponemos entonces de una forma rápida de recuperar el código fuente para depuración, y otros múltiples usos.

EL DEPURADOR/TRACEADOR

Tenemos aquí una poderosa ayuda para la corrección de programas complejos. Con el comando OPT podemos seleccionar el modo de operación del traceador entre cuatro (no excluyentes):

J: Modo JSR: Se para la ejecución al encontrar una instrucción RTS. Es muy útil para la depuración de subrutinas.

S: Modo STEP: Paso a paso. Se muestran los registros.

A: Modo ADRESS: Se muestra la dirección de cada instrucción que se ejecuta.

R: Modo REGISTRER: Se muestran los contenidos de los registros tras la ejecución de cada instrucción.

El comando DISP nos permitirá marcar una serie de puntos en los cuales se mostrará siempre el contenido de los registros; el comando LOC permite establecer además posiciones específicas de memoria cuyo contenido aparecerá junto con los registros.

CONCLUSIONES FINALES

No cabe duda de que MACHINE LIGHTNING es una de las mejores herramientas para el desarrollo de programas en código máquina que han pasado por nuestras manos.

A pesar de que sus autores lo enfocan principalmente al desarrollo de juegos, nada impide usarlo para cualquier otra aplicación que se nos ocurra.

Todas sus opciones son lo suficientemente claras como para que su uso sea sencillo, pero a la vez son lo suficientemente amplias y completas como para que nos permitan desarrollar eficazmente cualquier programa en código máquina.

Basic Lightning

BASIC LIGHTNING, el BASIC relámpago, es una potentísima versión del lenguaje BASIC, creada por la firma OASIS SOFTWARE y distribuida en España por SAKATI, S.A. En principio, los creadores de la versión la catalogan como una versión específicamente pensada para la creación de juegos a un nivel profesional.

Sin embargo, la cantidad y calidad de los comandos incluídos, nos hacen pensar en ella como algo más, como una versión de BASIC sumamente completa y bastante veloz que el usuario podrá utilizar en la confección de cuantos programas se le ocurran, tanto programas de juego como programas, digamos, serios.

Las facilidades incluidas en este BASIC abarcan desde comandos de carácter estructurado, hasta facilidades para el funcionamiento multitarea, pasando por innumerable comandos gráficos, de manejo del chip de sonido y de creación y manejo de sprites, entendidos estos últimos en un sentido más general que el de los sprites hardware del C-64.

PRESENTACION

El paquete de BASIC LIGHTNING, al igual que el resto de los paquetes LIGHTNING, viene presentado en una agradable caja de cartón que incluye dos cassettes con el programa y un completo manual. La versión que hemos tenido ocasión de probar incluía un manual en inglés, pero estamos seguros de que se proporcionará un manual en castellano en las versiones que se comercialicen en nuestro país. Como detalle curioso, el manual viene impreso en un papel de color rojo, lo que hace prácticamente imposible fotocopiarlo. Esto no es más que otra forma de poner las cosas difíciles para los amantes de la piratería.

LOS COMANDOS

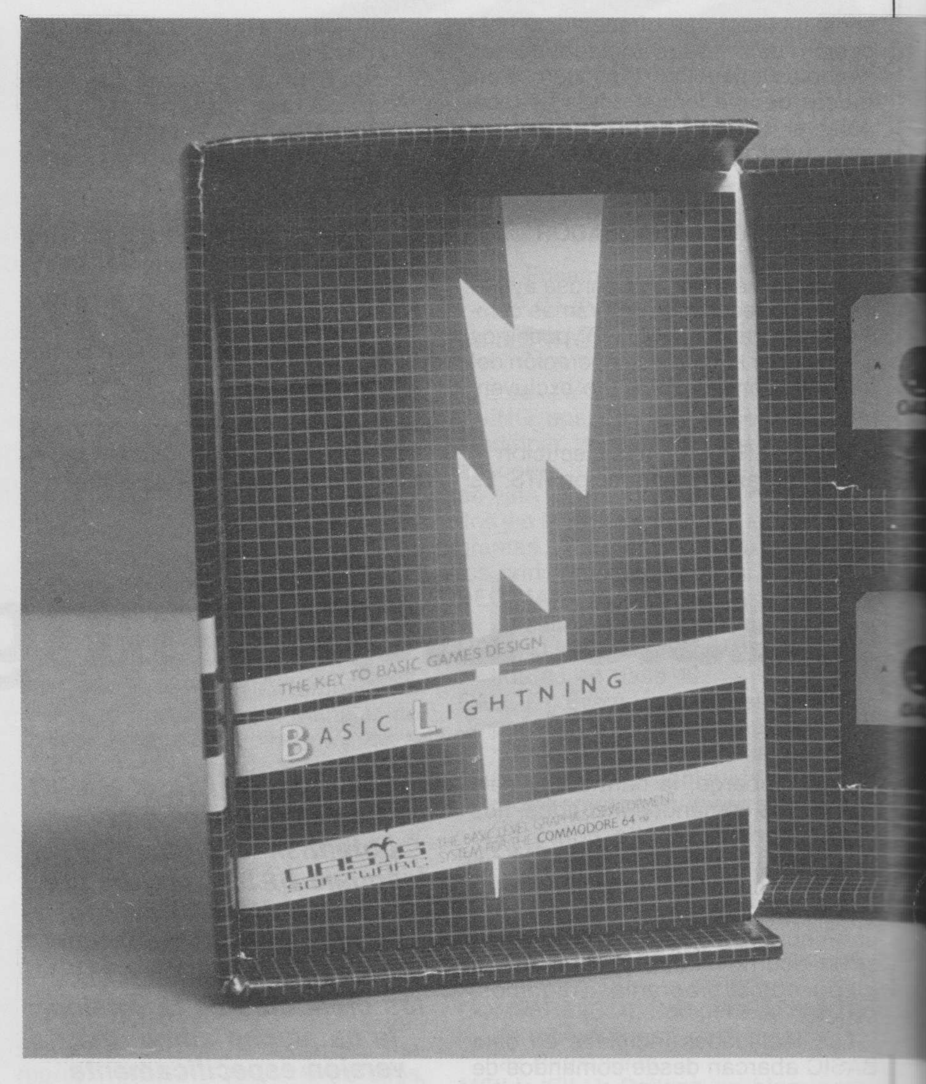
Vamos a entrar de lleno en los comandos que proporciona este BASIC LIGHTNING y que podemos dividir, siguiendo el manual, en seis apartados diferentes: programación estructurada, utilidades, gráficos, sonido, comandos multitarea y un programa para la generación de sprites.

PROGRAMACION ESTRUCTURADA

Los comandos de carácter estructurados son, en cierto modo, una adaptación de comandos y instrucciones características del lenguaje PASCAL, que han sido adaptados a las características del BASIC. Su función es la de permitir una programación más estructurada, más organizada, de forma que limitando en todos lo posible la utilización de GOTO se obtengan programas más comprensible, claros y sencillos de modificar.

Entre los comandos incluídos nos encontramos con los característicos IF-THEN-ELSE, extensión de la instrucción IF-THEN que permite tomar una vía de acción u otra según se cumpla o no la condición, REPEAT-UNTIL que permite repetir una acción determinada hasta que se cumpla una condición y WHILE-WEND

que se utiliza para repetir una acción mientras sea cierta una determinada condición. Además de estas instrucciones, está prevista la utilización de etiquetas, lo que permite asignar nombres simbólicos a determinados segmentos de programa, sin necesidad de tener que identificarlos por su número de línea. Por ejemplo, se puede asignar el nombre "cuadrado" a una subrutina que dibuje cuadrados y llamarla desde cualquier parte del programa mediante una instrucción GOSUB CUADRADO, sin preocuparse para nada de los números de línea. También se incluye, y esta es una de las características más importantes de la versión LIGHTNING, la posibilidad de utilización de procedimientos mediante la instrucción LIGHTNING, la posibilidad de utilización de procedimientos mediante la instrucción PROCEDURE. Los procedimientos son casi como subrutinas, pero ofrecen la posibilidad del paso de parámetros, es decir, que cuando se llame al procedimiento se le pasen unos valores con los que trabajar, valores que pueden ser distintos cada vez que se llama al procedimiento. Incluso es posible pasar como parámetro toda una matriz de valores. Además de los procedimientos se pueden definir y utilizar funciones, que no son más que segmentos de programa al modo de las subrutinas o procedimientos, pero que proporcionan un valor, el resultado de aplicar la función a los datos, al programa principal. Con todos estos comandos, la versión BASIC LIGHTNING se convierte en una interesante herramienta para la creacción de programas estructurados. La potencia de los comandos es mucha, solo hay que aprender a utilizarlos para empezar a apreciar todas sus posibilidades.



UTILIDADES

Las utilidades ofrecidas por BASIC LIGHTNING se presentan en forma de comandos que se pueden utilizar tanto en modo inmediato como desde programa. Se incluyen la conversión de números decimales en hexadecimales y viceversa, las instrucciones DEEK y DOKE que son iguales que los conocidos PEEK y POKE pero trabajan con dos bytes simultáneamente, las instrucciones DLOAD y DSAVE para la carga de programas desde la unidad de diskettes, la instrucción PULL que devuelve el último de los valores que se almacenó en la pila, y por último, la instrucción DIR que proporciona un listado de los programas incluídos en un diskette.

GRAFICOS

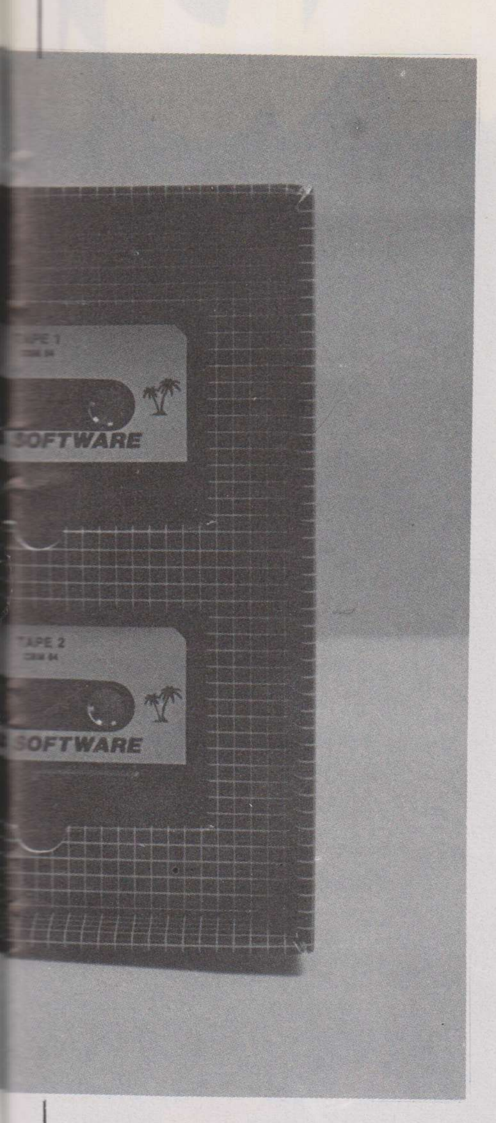
El apartado de gráficos es uno de los más completos y cuidados de esta versión BASIC LIGHTNING. Presenta la particularidad de estar orientado al manejo de un cierto tipo de sprites, distintos de los sprites hardware del C-64, y que se definen como caracteres gráficos definidos por el usuario de hasta 255 × 255 caracteres o bloques. Estos extraños sprites se manejas mediante 13 variables entre las que se incluyen el número, la fila o la columna de uno o dos sprites, el ancho y el alto de la ventana en la que se ha definido el sprite y la fila y la columna para la detección de las colisiones. Además de estas 13 variables, existen hasta 39 comandos para el movimiento de los bloques que constituyen los sprites. Entres estas 39 funciones se incluyen desde el simple movimiento de sprites hasta la realización de operaciones AND, OR y OR exclusivo entre diferentes blo-

THE KEY TO BASIC GAMES DESIGN

BASIC LIGHTNING

OASIS SOFTWARE

THE BASIC LEVEL GRAPHICS DEVELOPMENT SYSTEM FOR THE COMMODORE 64 ™



ques. La combinación de las instrucciones con los distintos valores de los trece parámetros proporciona un buen montón de posibilidades de trabajo con todo tipo de gráficos de alta resolución. En principio puede resultar un poco complicado este sistema de sprites, pero en cuanto uno de pone a trabajar con ellos, se aprecia un enorme potencial con una muy buena velocidad de respuesta y todo mediante una serie de comandos BASIC, sin necesidad de recurrir a POKES a difíciles posiciones de memoria.

Pero hay mucho más. Además de estas facilidades con bloques gráficos, hay una serie de comandos como PLOT, DRAW, BOX y POLY que permiten el trabajo en alta resolución con puntos, líneas, rectángulos y polígonos de cualesquiera forma, color y dimensiones. Por ejemplo, la instrucción POLY 0, 32, 32, 32, 32, 5, 0 permitirá dibujar un pentágono en la esquina izquierda de la pantalla a velocidades vertiginosas y en alta resolución. Hay 18 comando para el control de scrolling en el interior de las ventanas en las que se define cada sprite. Por ejemplo se incluyen las instrucciones SCROLL y WRAP que permiten hacer dos tipos de scrolling, con SCROLL, los gráficos que desaparecen por uno de los extremos de la ventana no vuelven a aparecer, con WRAP vuelven a surgir, a medida que desaparecen por un extremo, por el extremo opuesto. Además, en todos estos comandos de scrolling, es posible elegir de cuántos en cuantos pixels se va a llevar a cabo el desplazamiento y cuales van a ser las dimensiones de la ventana, bloque gráfico o sprite en el que va tener lugar.

Otra de las interesante posibilidades de trabajo con los sprites es la de las transformaciones de los mismos. Existe la posibilidad de reflejar el sprite respecto de una línea horizontal o vertical mediante las instrucciones FLIP y MIR. SPIN permite hacer girar un sprite o una ventana un ángulo determinado. Por último, XPAND permite expandir una ventana ya sea en la dirección horizontal o en la vertical.

Otras de las opciones ofrecidas dentro del apartado de los gráficos son las del manejo de los sprites hardware del Commodore, pero eso sí, empleando única y exclusivamente instrucciones BASIC y olvidándose de los tediosos POKES o del lenguaje máquina.

Por último, dentro de este apartado se incluyen comandos que permiten leer el teclado, leer el joystick, leer el lápiz óptico, incluir caracteres en la pantalla de alta resolución y modificar los colores del borde, de la pantalla y de los caracteres.

En definitiva, BASIC LIGHTNING no permite hacer nada que no pudiera hacer un programador experto en gráficos, pero se encarga de facilitar las cosas al máximo. La utilización adecuada de los comandos permitirá a cualquier usuario medio la realización de no importa que tarea gráfica en alta resolución, y ello simplemente utilizando comandos BASIC. La idea es muy atractiva ¿no?.

SONIDO

Para generar sonidos por cualquiera de las tres voces que proporciona el chip SID del Commodore 64 hace falta controlar cuatro parámetros que son el volumen, la frecuenca del sonido, la envolvente y la forma de onda. BASIC LIGHTNING proporciona comandos para el control directo de estos parámetros. VOLUME permite controlar el volumen entre los valores de cero y quince, FRQ la frecuencia, ADSR permite controlar independientemente los valores de ataque, caída, sostenimiento y relajación del sonido y, por último, los comandos TRI, SAW, PULSE y NOISE permiten el control de la forma de onda. Solo con estos comandos ya resulta posible generar una buena cantidad de sonidos, pero sin embargo, las posibilidades del chip SID van mucho más alla, y BASIC LIGHTNING incluye comandos para estas otras posibilidades. Por ejemplo, el comando FILTER permite el control de los filtros de cada una de las voces, RING controla la modula-

SUMME

Para COMM

EPYX
COMPUTER SOFTWARE

Strategy Games for the Action-Game Player



Fabricado y distribuido bajo licencia por:
COMPULOGICAL S.A.

- Ceremonia de apertura y
  entrega de premios.
- Ocho competiciones depor
  rentes.
  - Salto de pértiga.
  - Salto de trampolín.
  - 4 × 400 metros relevos.
  - 100 metros lisos.
  - Gimnasia.
  - Natación estilo libre rele
  - Natación 100 metros libr
  - Tiro al blanco.
- Compita contra el ordena
  o contra sus amigos o fami
- Juego individual o en
  equipo.
- Control joystick.
- Uno o más jugadores.



Solicítela a: **INFODIS C/ Bravo Murillo, 377 - 5.° A - 28020 MADRID**

CUPON DE PEDIDO

Sí, envíenme al precio de 2.300 ptas. ... ejs. de SUMMER GAMES.

El importe lo abonaré: Contra reembolso ☐ Con mi tarjeta de crédito ☐

Adjunto cheque ☐ American Express ☐ Visa ☐ Interbank ☐

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Fecha de caducidad ____

NOMBRE ____

DIRECCION ____

CIUDAD ____

PROVINCIA ____

Sin gastos de envío

GAMES

SERIE DIAMANTE

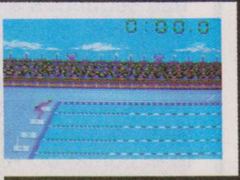
E 64 y 128

Precio 2.300 Ptas.

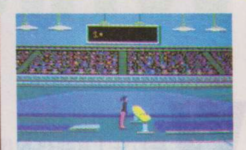
emonia de

s dife-









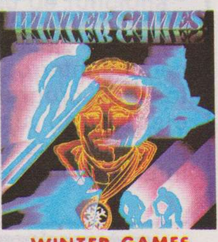
Comercializado por infodis, s.a.





SUMMER GAMES





WINTER GAMES

TA A LA VENTA

ción en anillo y SYNC la sincronización.

COMANDOS MULTITAREA

BASIC LIGHTNING ofrece la interesante posibilidad de trabajar en modo multitarea, es decir, con varias tareas o partes de un mismo programa funcionando simultáneamente. Para ello se emplea la técnica denominada tiempo de CPU (BASIC LIGHTNING asigna tiempos de 1/20 segundos a cada una de las tareas) por riguroso turno. Antes de entrar a funcionar en modo multitarea hay que reservar espacio para las variables de cada una de las tareas, lo que se lleva a cabo mediante el comando ALLOCATE. Después de ello, para llamar a una de las tereas no hay má que utilizar el comando TASK, con los parámetros que identifiquen a la tarea concreta. Por último, el comando HALT permite detener tareas. De este modo se pueden programar tareas distintas de una forma mucho más sencilla. Por ejemplo se podría pensar en una tarea que se encargara de leer caracteres de entrada mientra simultáneamente otra tarea se encarga de ordenar las palabras que se vayan formando con dichos caracteres y una tercera se ocupara de la lectura de datos desde un cassette.

EL GENERADOR DE SPRITES

Como complemento al conjunto de utilidades y comandos de BASIC LIGHTNING se incluye un programa para la generación de sprites, tanto de los sprites hardware de Commodore como de los sprites software específicos del programa. El programa incluye cinco modos de funcionamiento a los que se accede a través de las tecla de función. Los modos se encargan de la generación de sprites en dos colores o en cuatro colores, de la carga y el almacenamiento de los mismo en diskette y el movimiento de estos por la pantalla. El manejo del programa se hace de una forma interactiva, trabajando sobre la pantalla, por loq ue su utilización resulta sumamente agradable.

CONCLUSION

BASIC LIGHTNING es un completo paquete pensado en un principio para la generación y el trabajo con gráficos, pero que incluyen capacidades tan interesantes como comandos de carácter estructurado y multitarea.

Para los que quieran explorar las capacidades gráficas y de todo tipo del Commodore 64 utilizando comandos BASIC, el programa resulta más que recomendable. □

White Lightning

WHITE LIGHTNING es el tercero de la serie de programas LIGHTINING desarrollados por la firma OASIS SOFTWARE y comercializados en nuestro país por SAKATI, S.A. El paquete se compone de dos partes diferentes, la primera de ellas es una versión del lenguaje FORTH mientras que la segunda la compone un sublenguaje denominado IDEAL. Este sublenguaje contiene hasta 100 palabras que pueden mezclarse con las instrucciones FORTH y presenta la notable característica de que su ejecución puede llevarse a cabo mediante interrupciones. Esto quiere decir que mientras un programa se está ejecutando en modo normal, otro programa se puede ejecutar cuento se producen las interrupciones. Es como si los dos programas se estuvieran ejecutando simultáneamente. El programa que se ejecuta mediante interrupciones recibirá atención por

THE KEY TO PROFESSIONAL GAMES DESIGN
WHITE LIGHTNING
THE HIGH LEVEL GRAPHICS DEVELOPMENT SYSTEM FOR THE COMMODORE 64
OASIS SOFTWARE

parte del procesador sólo cada 1/60 de segundo, pero esto es más que suficiene para que el procesador y el sistema operativo se encuentren en cierto modo descargados de trabajo. El resultado es un funcionamiento más veloz y en el que más cosas se pueden hacer al mismo tiempo.

FORTH

Sobre la versión de lenguaje FORTH no nos extenderemos. Diremos sólo que se ajusta a las normas del Fig-Forth estandar, pero con las cien extensiones al vocabulario Forth mencionadas. En conjunto la versión resulta completa y rápida y



sólo por ella puede decirse que merece la pena este paquete pero es que además nos encontramos con el sublenguaje IDEAL.

EL SUBLENGUAJE IDEAL

El sublenguaje IDEAL, cuyas siglas corresponden a las de Lenguaje de Animación Extensible Dirigido por Interrupciones, está constituido por un conjunto de comandos y un conjunto de utilidades que se añaden a las de la versión FORTH. Estos comandos están fundamentalmente orientados al manejo de los gráficos y del sonido. Los gráficos se desarrollan mediante una serie de sprites software, a los que hay que añadir los sprites hardware del Commodore. Estos sprites presentan las mismas características que los de la versión BASIC LIGHTNING, por lo que no volveremos a comentarlas. Lo mismo podemos decir con respecto a los comandos para generación de sonidos. Se trata de un conjunto de comandos, los mismos que en la versión BASIC LIGHTNING que permititán controlar todos los registros del chip SID de una forma clara, asequible y rápida.

LAS INTERRUPCIONES

Se trata de una de las características más sobresalientes de este paquete y consiste en la posibilidad de ejecutar toda una serie de palabras mediante la técnica de las interrupciones. Es decir, cada cierto tiempo, el programa que se está ejecutando se interrumpe y comienza y comienza la ejecución de alguna de las palabras del sublenguaje. Terminada la ejecución de esta, se reanuda la del programa principal. Este tratamiento mediante interrupciones puede aplicarse a cualquiera de los comandos o palabras, tanto si son de gráficos como si se refieren al sonido. El efecto es el de un incremento en la velocidad de ejecución total de los programas unido a una mayor sencillez de programación, especialmente por lo que se refiere a tareas complejas que pueden simultáneamente.

CONCLUSIONES

Al igual que las versiones BASIC LIGHTNING y MACHINE LIGHTNING, esta versión del lenguaje Forth, WHITE LIGHTNING, ofrece una gran cantidad de comandos para una programación más eficiente y para un aprovechamiento completo de las posibilidades del C-64. A una gran velocidad de funcionamiento se une una enorme sencillez de manejo y una increíble flexibilidad de utilización. En definitiva nos encontramos ante tres poderosísimas herramientas para el desarrollo de programas, tanto a nivel de aficionado como de profesional. □

Sprites múltip



Más grandes, más bonitos, más perfectos...

¿Eres capaz? Inténtalo. ¡Tú puedes!

En la memoria de tu ordenador y en un mismo programa puedes guardar más de cien Sprites, de los que puedes disponer en cualquier momento. Para conseguirlo lee atentamente, pero antes recuerda que cada Sprite se compone de 63 números (un Sprite ocupa 63 bytes) que introducimos en siete sentencias DATA, para luego ser descifrados por un bucle de lectura que los retendrá en memoria.

Esto lo hacíamos con un solo Sprite, ahora para introducir varios, lo que tenemos que hacer es colocarlos para que formen bloques de 64 bytes.



Si fuiste capaz de crear Sprites monocolores y multicolores, ahora tienes la oportunidad de diseñar otros duendes, más grandes, más bonitos, más perfectos.
Un Sprite múltiple es un diseño que se compone de varios Sprites, los cuales al estar juntos forman una gran figura, un duende que puedes mover por la pantalla, darle vida y hacerle aparecer y desaparecer a tu gusto.
Destruye tu nave en pedazos; haz que tu coche pierda las ruedas, dibuja el anagrama de tu empresa o sorprende a tus amigos haciendo aparecer tu mascota en cualquier programa; y ya sabes, si algo te falla, vuelve a leer el artículo y te aseguro que encontrarás la solución.

Para conseguirlo sólo tienes que añadir a cada Sprite que introduzcas un 0 al final, de manera que tenga 64 números y nos forme un bloque.

Cambiar entonces el bucle de lectura para que nos lea tantos bloques de 64 números como Sprites deseamos introducir.

```
FOR J = 0 TO X * 64 - 1 : READ Q :
POKE 12288 + J, Q : NEXT
```

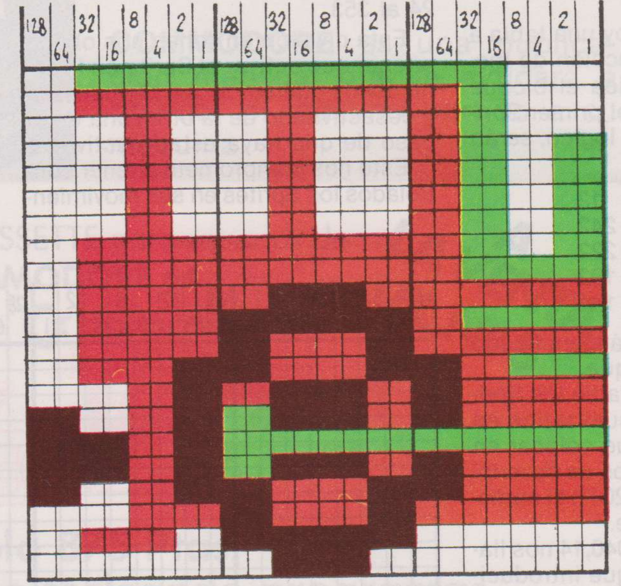
(siendo X el número de Sprites que vayamos a introducir)

Ahora vamos a estudiar la segunda parte del bucle de lectura, que son las direcciones de memoria.

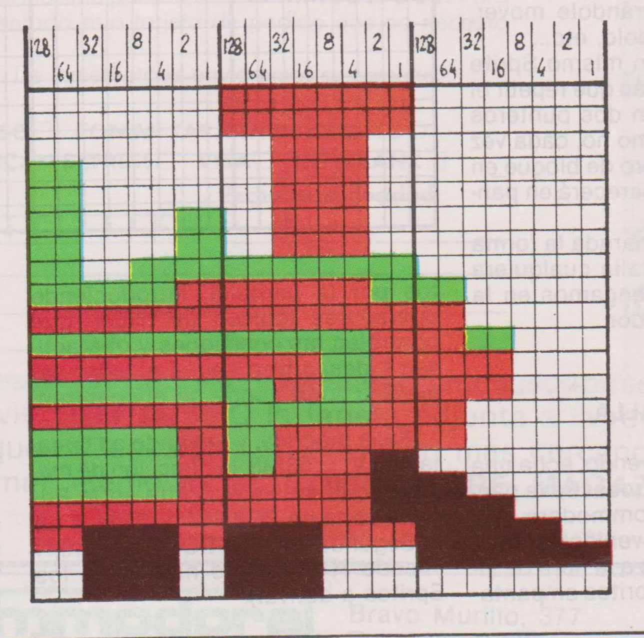
Habitualmente los principiantes utilizan la dirección 832 en adelante, donde sólo caben 3 Sprites; para superar esta cantidad sería conveniente utilizar otras, yo te recomiendo éstas:

Hasta 14 Sprites
Hasta 34 Sprites
Hasta 64 Sprites
POKE 15488 en adelante
POKE 14208 en adelante
POKE 12288 en adelante

Hay otras direcciones posibles en el segundo bloque de la RAM de tu Commodore, exactamente a patir de 58368 que puede contener hasta 112 Sprites, pero no te aconsejo hasta que no domines bien las direcciones de memoria de tu ordenador.



21	85	80
42	170	144
8	32	144
8	32	145
8	32	145
8	32	145
8	32	145
42	170	145
42	170	149
42	190	170
42	255	149
42	235	170
43	235	229
11	190	234
203	126	234
251	85	85
251	126	234
203	235	234
10	235	170
40	255	0
0	60	0



0	170	0
0	170	0
0	40	0
64	40	0
64	40	0
65	40	0
65	40	0
69	85	0
86	169	64
170	170	128
85	86	144
170	166	160
85	102	160
170	86	128
86	106	128
86	107	0
86	171	192
170	171	240
190	250	252
60	240	255
60	240	0

Con esto ya tenemos comprendido y solucionado el bucle de lectura y las direcciones de memoria donde van a permanecer tus Sprites.

VISUALIZA TUS SPRITES

Ahora que ya tienes tus Sprites retenidos en bloques de 64 bytes, necesitas saber en qué bloque está cada uno de ellos para poder visualizarlos en pantalla.

A continuación te doy una tabla a emplear, donde la dirección de memoria utilizada te indica el bloque donde está guardado el primer Sprite; los demás, como es lógico, serán los sucesivos:

POKE 832 bloque 13
POKE 15488 bloque 242
POKE 14208 bloque 222
POKE 12288 bloque 192
POKE 58368 bloque 144

Así cada Sprite estará identificado por un número de bloque.

Para hacer la llamada de los Sprites cuando quieras visualizarlos en pantalla sólo tienes que colocar en uno de los ocho punteros de descripción (POKE 2040 hasta 2047) el número de bloque del Sprite.

Por ejemplo POKE 2040,14 nos llama al segundo Sprite que introducimos en la dirección POKE 832, tomando el valor del Sprite 0 (nivel 1) en el programa, lográndole mover, dar color, agrandándolo, etc...

Si deseas tener un mismo Sprite dos veces, sólo tendrás que repetir el número de bloque en dos punteros de descripción, y cómo no, cada vez que cambies el número de bloque en un mismo puntero aparecerá en pantalla el nuevo Sprite.

Con esto queda aclarada la forma de visualizar en pantalla cualquiera de los Sprites que tengamos en la memoria del Ordenador.

CONOCE TU PANTALLA

Antes de seguir leyendo, echa una ojeada al gráfico que muestra la pantalla visible de un Commodore.

Uno de los inconvenientes más grandes que tenemos a la hora de situar y mover varios Sprites en pantalla está motivado por los límites que nos establece la dirección de memoria del eje X de cualquiera de nuestros Sprites.

Dicho más claramente, al activar un Sprite, el máximo valor que permite el eje X y el eje Y es de 255. En el eje Y no nos afecta, pues las coordenadas visibles están entre 50 y 249. Pero el eje X nos obliga a partir la pantalla en dos zonas de 255 puntos cada una donde sólo será visible del 24 al 353.

Esta división en zonas nos obliga a tener que activar el Sprite en la zona donde queremos que permanezca y desactivando de la otra zona en el caso de que haya estado activado.

Esto nos compromete a tener controlados los Sprites en sus movimientos por la pantalla, introduciendo operadores lógicos (IF-THEN) que controlen las posiciones y que activen y desactiven según el valor del eje X o zona donde se encuentren.

En la práctica, y teniendo en cuenta que V = 53248, la dirección de memoria que activa el Sprite en la Primera zona es POKE V + 21,H, y en la Segunda zona es POKE V + 16,H (siendo H el nivel o niveles de los Sprites a activar).

Cuadro de niveles:

Sprite	0	nivel	1
Sprite	1	nivel	2
Sprite	2	nivel	4
Sprite	3	nivel	8
Sprite	4	nivel	16
Sprite	5	nivel	32
Sprite	6	nivel	64
Sprite	7	nivel	128

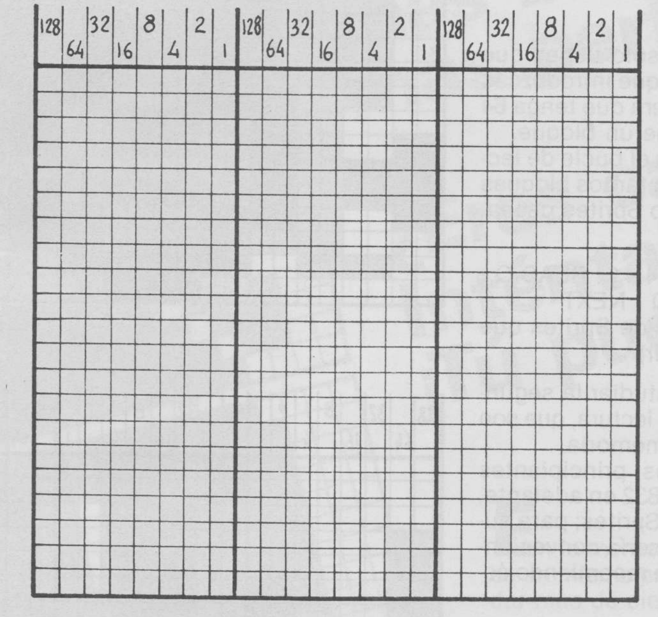
Con esto queda aclarado el gráfico de pantalla de tu Commodore y ya no nos queda nada más que empezar el diseño de nuestro Sprite múltiple.

DISEÑO DE UN SPRITE MULTIPLE

El diseño de un Sprite múltiple es una composición de varios Sprites que forman un gran dibujo en Alta resolución.

Un Sprite múltiple se puede componer de Sprites monocolores y multicolores indistintamente, al igual que superponerse unos con otros de cualquier forma.

Incluyo una gráfica para confeccionar Sprites, de la que tendrás que sacar fotocopias o confeccionar las tuyas propias con papel cuadriculado.







LA REVISTA IMPRESCINDIBLE PARA LOS USUARIOS DE LOS ORDENADORES PERSONALES **COMMODORE.**

COMMODORE es una publicación mensual que le ayudará a obtener el máximo partido a su máquina.

CONOZCA LAS VENTAJAS DE SUSCRIBIRSE A



Sensacional Oferta de Suscripción





ADEMAS, le hacemos un 25 % DE DESCUENTO

sobre el precio real de suscripción (12 números)

VALOR REAL DE SUSCRIPCION	OFERTA ESPECIAL DE SUSCRIPCION	USTED AHORRA
~~3.600~~ PTAS.	2.700 PTAS.	900 PTAS.

APROVECHE AHORA esta oportunidad irrepetible para suscribirse a **COMMODORE MAGAZINE**. Envie **HOY MISMO** la tarjeta adjunta a la revista, que no necesita sobre ni franqueo. Deposítela en el buzón más cercano. Inmediatamente recibirá su primer ejemplar de **COMMODORE MAGAZINE** más el **REGALO**.



Bravo Murillo, 377
Tel. 733 79 69
28020 MADRID

Ya puedes ponerte a dibujar, pero no olvides la diferencia entre un diseño monocolor y otro multicolor.

Para comprender mejor todo esto, es interesante que repases el artículo sobre Sprites multicolores que publicamos en el número anterior.

Nuestro programa ejemplo, como puedes ver, es una máquina de tren que se compone de dos Sprites. No he querido hacer algún Sprite múltiple más grande porque lo interesante es que desarrolles tu imaginación.

Por ahora sólo te propongo que con los seis Sprites que te quedan intentes colocar vagones a la máquina; estoy seguro que lo conseguirás, pero por ahora atiende al programa ejemplo.

Comenzamos en la línea 150 porque en las primeras 140 líneas tienes que introducir los DATA de tu máquina (cuando introduzcas vagones tendrás que comenzar más adelante).

SPRITE MULTIPLE MULTICOLOR «MAQUINA DE TREN»

Blanco	- Fondo
Verde	- 1.º Color Común
Negro	- 2.º Color Común
Rojo	- Sprite

```
DATA  21,85,80,42,170,144,8,32,144
DATA  8,32,145,8,32,145,8,32,145
DATA  8,32,145,42,170,145,42,170,149
DATA  42,190,170,42,255,149,42,235,170
DATA  43,235,229,11,190,234,203,126,234
DATA  251,85,85,251,126,234,203,235,234
DATA  10,235,170,40,255,0,0,60,0

DATA  0,170,0,0,170,0,0,40,0
DATA  64,40,0,64,40,0,65,40,0
DATA  65,40,0,69,85,0,86,169,64
DATA  170,170,128,85,86,144,170,166,160
DATA  85,102,160,170,86,128,86,106,128
DATA  86,107,0,86,171,192,170,171,240
DATA  190,250,252,60,240,255,60,240,0
```

La primera línea es el bucle de lectura de los 2 Sprites, y los guardamos en la dirección de memoria 15488, donde pueden entrar hasta 14 Sprites:

```
150 FOR J = 0 TO 2 * 64 — 1 : READ Q : POKE 15488 + J, Q : NEXT
```

Para limpiar la pantalla y ponerla de color verde:

```
160 PRIN «[CLS]» : POKE 53281,5
```

Dibujamos una línea en la pantalla que nos servirá de vía del tren:

```
170 FOR A = 0 TO 39 : POKE 1864 + A,64 : POKE 56136 + A,6 : NEXT
```

Damos valor a V para más comodidad:

```
180 V = 53248
```

Sacamos los Sprites de los bloques 242 y 243 y les asignamos un punterro de descripción a cada uno:

```
190 PCKE 2040 , 242 : POKE 2041 , 243
```

Activamos el modo multicolor para los Sprites 0 y 1 (niveles 1 y 2):

```
200 POKE V + 28,3
```

Ponemos el Primer Color Común en color Cian y el Segundo Color Común en negro:

```
210 POKE V + 37,3 : POKE V + 38,0
```

Ponemos el color propio de cada Sprite en gris oscuro:

```
220 POKE V + 39,11 : POKE V + 40,11
```

Dejamos el tamaño pequeño de los Sprites:

```
230 POKE V + 23,0 : POKE V + 29,0
```

Activamos la zona de pantalla donde van a aparecer, desactivando la otra zona:

```
240 POKE V + 21,3 : POKE V + 16,0
```

Bucle de movimiento de los dos Sprites (valores de X e Y):

```
250 FOR X2 = 0 TO 290 : X1 = X2 + 24 : Y = 200
```

Si X2 es mayor que 255 activa el Sprite 0 en la segunda zona, y lo mismo con X1:

```
266 IF X2 > 255 THEN POKE V + 16 , (PEEK (V + 16) AND 254) OR (INT (X2/56) * 1)
270 IF X1 > 255 THEN POKE V + 16 , (PEEK (V + 16) AND 254) OR (INT (X1/256) * 2)
```

Demos la situación a los Sprites según el valor de X2 y X1 (eje X):

```
280 POKE V + 0 , X2 AND 255
290 POKE V + 2 , X1 AND 255
```

Demos la situación a los Sprites según el valor de Y (eje Y):

```
300 POKE V + 1 , Y
310 POKE V + 3 , Y
```

Cierra el bucle:

```
320 NEXT
```

Has visto qué fácil, ahora sólo tienes que echarle un poco de imaginación y hacer un programa que coloque vagones a tu tren.

Y por último, unos pocos consejos:

— Si quieres más rapidez, ya sabes que tienes que utilizar código máquina.

— Recuerda que entre los Sprites hay unas prioridades absolutas e inalterables de los primeros sobre los últimos.

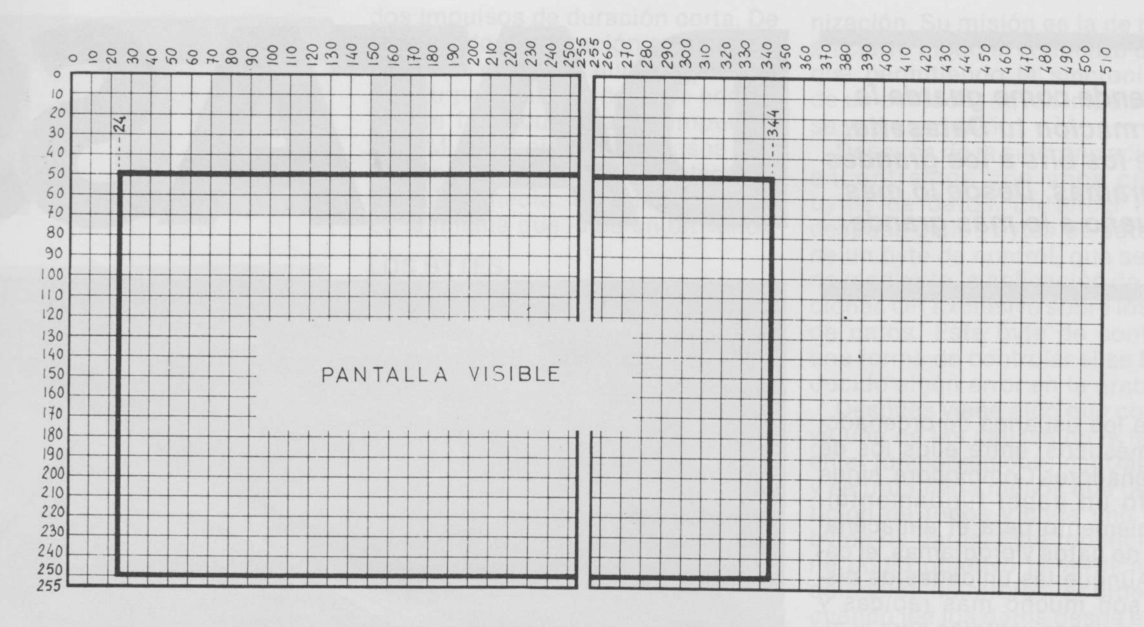
— Intenta aprovechar la vibración entre Sprites como parte de animación de tus diseños.

— Si haces programas muy largos con Sprites, puedes pisar la zona de memoria donde están guardados, apreciándose éstos con formas extrañas en tu pantalla.

— Si no comprendes algo, puede ser que no hayas leído el artículo de la revista anterior que trataba sobre los Sprites multicolores.

— En el próximo número estudiaremos la forma de mover los Sprites en horizontal, vertical, diagonal, círculo, elipse, haciendo ochos, etc...

Rubén



GRAFICA DE COORDENADAS DE LA PANTALLA VISIBLE

```
10 DATA 21,85,80,42,170,144,8,32,144
20 DATA 8,32,145,8,32,145,8,32,145
30 DATA 8,32,145,42,170,145,42,170,149
40 DATA 42,190,170,42,255,149,42,235,170
50 DATA 43,235,229,11,190,234,203,126,234
60 DATA 251,85,85,251,126,234,203,235,234
70 DATA 10,235,170,40,255,0,0,60,0,0
80 DATA 0,170,0,0,170,0,0,40,0
90 DATA 64,40,0,64,40,0,65,40,0
100 DATA 65,40,0,69,85,0,86,169,64
110 DATA 170,170,128,85,86,144,170,166,160
120 DATA 85,102,160,170,86,128,86,106,128
130 DATA 86,107,0,86,171,192,170,171,240
140 DATA 190,250,252,60,240,255,60,240,0,0
150 FORJ=0TO2*64-1:READQ:POKE15488+J,Q:NEXT
160 PRINT"♥":POKE 53281,5
170 FORA=0TO39:POKE1864+A,64:POKE56136+A,6:NEXT
180 V=53248
190 POKE2040,242:POKE2041,243
200 POKE V+28,3
210 POKE V+37,3:POKEV+38,0
220 POKE V+39,11:POKEV+40,11
230 POKE V+23,0:POKEV+29,0
240 POKE V+21,3:POKEV+16,0
250 FOR X2=0 TO 290: X1=X2+24:Y=200
260 IF X2>255 THEN POKEV+16,(PEEK(V+16) AND 254) OR (INT(X2/56)*1)
270 IF X1>255 THEN POKEV+16,(PEEK(V+16) AND 254) OR (INT(X1/256)*2)
280 POKEV+0,X2 AND 255
290 POKEV+2,X1 AND 255
300 POKEV+1,Y
310 POKEV+3,Y
320 NEXT
```

DATASS

Aprende como guarda la información tu Datasette. Desde los bits a los grandes programas. Desde lo más pequeño a lo más grande.



Entre los usuarios de ordenadores domésticos, entre ellos los de los ordenadores Commodore, sigue jugando un papel fundamental, como elemento para el almacenamiento de datos y programas, el casette. Aunque las unidades de diskettes son mucho más rápidas y ofrecen muchas ventajas para el almacenamiento de información, presentan, al mismo tiempo, una desventaja fundamental; su todavía elevado precio. Así pues, por el momento, las unidades de casettes y las cintas de casette siguen siendo el medio más universal de almacenamiento de la información que generan los microodenadores. Por ello nos ha parecido interesante echar un vistazo más de cerca a la forma en la que la información, los datos y programas, quedan almacenados en los casettes y concretamete en la unidad a casette de Commodore; el Datasette.

DE LOS PROGRAMAS A LOS BITS

Lo que nosótros normalmente decidimos grabar en el Datasette son programas. Estos estan constituidos por un conjunto de instrucciones, las cuales, a su vez, no son más que un conjunto de caracteres que a su vez vienen representados en el interior del ordenador por bytes, cada uno de los cuales no es más que un conjunto de ocho bits. Así pues, al nivel más alto de resolución, lo que se almacena en la cinta que colocamos en el Datasette son bits, «unos» y «ceros», estados altos o bajos de tensión. Sin embargo, el Datasette no se queda a este nivel sino que codifica cada uno de los estados, el «uno» y el «cero», mediante una serie de impulsos de diferente duración que le sirven además de codificar otros tipos de información sobre lo que va almacenando en la cita. Estos impulsos eléctricos son de tres tipos que se diferencian sólo por su duración.

Impulsos de corta duración (176 microsegundos). Los llamaremos impulsos C.

Impulsos de duración media (256 microsegundos). Serán los impulsos M.

Impulsos de larga duración (336 microsegundos). A estos los llamaremos impulsos L.

Con estos tres tipos de impulsos se almacena toda la información en el Datasette.

LOS BITS

Los dos bits «cero» y «uno» se codifican mediante los impulsos antes mencionados de la siguiente forma.

Al bir «uno» le corresponde la secuencia MMCC, esto es, dos impulsos de duración media seguidos de



dos impulsos de duración corta. De este modo, la duración total de un bit «uno» es de 864 microsegundos. Por su parte al bit «cero» le corresponde la secuencia de impulsos CCMM, es decir, dos impulsos cortos seguidos de dos impulsos de duración media. Su duración total es la misma que la de un bit «uno».

LOS BYTES

Pero los bits no se graban todos uno a continuación de otro sin ningún orden ni concierto, sino que se agrupan de ocho en ocho formando bytes. El Datasette tiene que tener alguna forma de identificar donde comienza el primer bit de cada byte. Aquí es donde intervienen los impulsos que hemos denominado de larga duración o impulsos L. Antes de cada bloque de ocho bits, de la ristra de impulsos C y M que constituyen la combinación de ocho uno y ceros, el Datasette incluye la secuencia de impulsos LLMM como una marca que señala el comienzo del byte. Esta secuencia es un poco más larga que la que constituye un bit ya que su duración es de 1184 microsegundos. De este modo nos encontramos la información grabada en el Datasette en forma de una serie de caracteres constituidos cada uno de ellos por una marca de byte seguida de ocho bits. Pero además, después del último bit de los ocho que forman el byte, el Datasette incluye otro bit denominado bit de parida, que vale a modo de comprobande de que no ha habido errores, cuyo valor cero o uno depende de los valores de cada uno de los ocho bits. De este modo nos encontramos con caracteres con una duración total de 8960 microsegundos.

LOS BLOQUES

Si nos fijamos ahora en un nivel superior en la forma de grabación de los datos, veremos como en realidad los bytes se estructuran en una unidades superiores que se denominan bloques. Un bloque está formado por un conjunto de bytes, tanto de datos como de control, organizados de la siguiente manera. En primer lugar nos encontramos con unos cuantos bytes de sincronización. Su misión es la de marcar en cierto modo el comienzo del bloque permitiendo la sincronización de un contador de los impulsos que se van a producir.

Después de la sincronización nos encontramos con unos cuantos bytes de datos. En tercer lugar y después de los bytes de datos, viene un byte de control, que se obtiene mediante la aplicación de operaciones OR exclusivo sobre los bytes de datos. Este byte de control es una forma de controlar si se ha producido algún error en la grabacíon.

Después viene algo que constituye una de las causas de la lentitud en el funcionamiento del Datasette y se trata nada menos que de la petición de todos y cada uno de los bytes de datos. Esta repetición tiene como objetivo la verificación y detección de errores. El ordenador, cuando lee los datos desde el Datasette, los lee por duplicado y los compara. Si las dos copias no son iguales significa que ha habido algún error lo que se refleja en la pantalla a través del conocido LOAD ERROR. Después de la repetición de los datos viene la repetición del byte de control. Por último, un último byte indica el final del bloque.

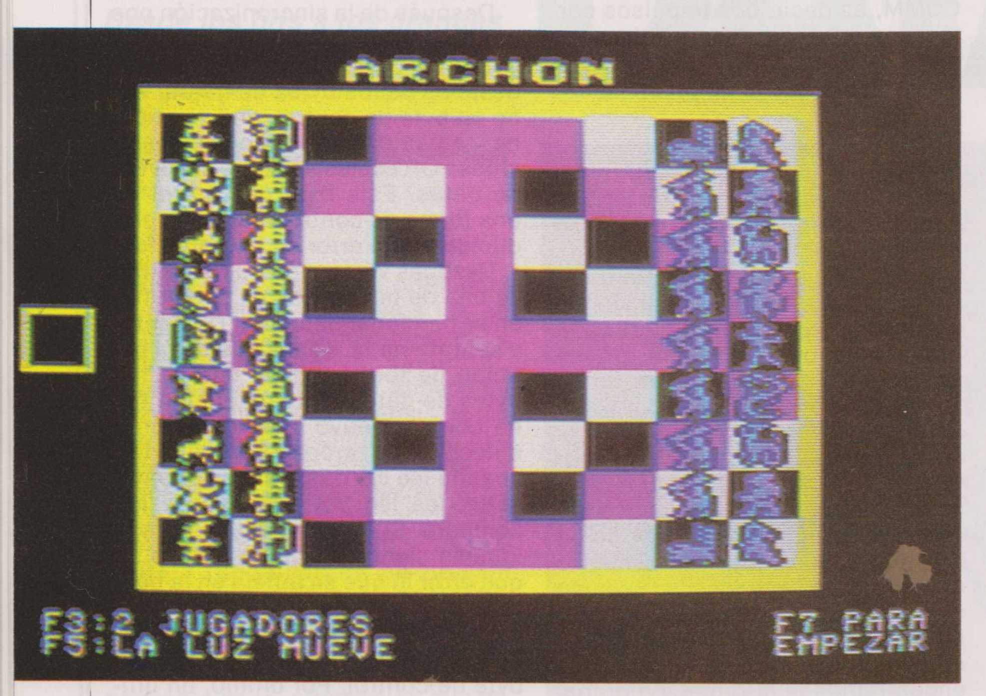
FICHEROS EN GENERAL

Por último, si consideramos la grabación de información desde el nivel más alto, nos encontramos con los ficheros que pueden ser programas o ficheros de datos. Todos los ficheros se almacenan en el Datasette bajo el siguiente formato. En primer lugar nos encontramos con una cabecera. Esta cabecera está formada por algunos bloques que contienen información sobre el fichero. Después de la cabecera vienen los bloques de datos propiamente dichos, es decir los bloques que contienen los datos del fichero o las instrucciones del programa.

Después de estos bloques de datos y según los casos aparece un último bloque que se conoce con el nombre de bloque EOT (End Of Tape o Final de la Cinta) y sirve para indicarle al ordenador que se ha terminado la cinta o que a partir de la marca no podrá encontrar más información. □

Archon





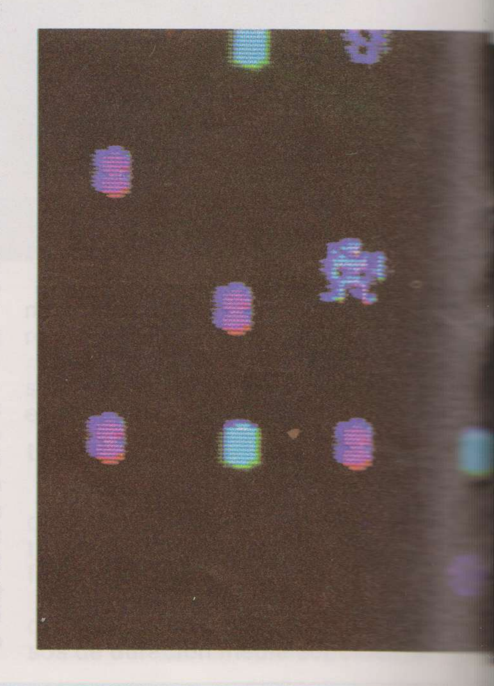
Distribuido en España por la firma DRO SOFT, se trata de uno de los juegos más originales y simpáticos que hemos visto. ARCHON combina adecuadamente elementos de un juego de tablero, con batallas entre sprites al clásico estilo de los marcianitos. En esencia, el juego es una batalla entre las fuerzas de la luz y la oscuridad, que se desarrolla sobre un tablero de 9 × 9 cuadros. Hay cuadros blancos y negros, y cuadros que cambian de color de forma cíclica en el transcurso de la partida. Al comienzo del juego se puede elegir entre la opción a dos jugadores, uno con cada bando, y la opción de luchar con uno de los bandos contra el C-64, que controlará el bando contrario. Una vez que comienza la partida, los jugadores pueden ir moviendo sus piezas por turno a través del tablero. Las piezas, que son distintas para cada uno de los bandos, son simpáticas representaciones de seres mitológicos o de fábula. Así, en el bando de la luz nos encontramos con unicornios, ave fenix, valkiria, golems.... seres que se enfrentarán con los ogros, dragones, trolls... del bando de la oscuridad. Cada uno de estos seres posee unas características propias como su capacidad de movimiento, de disparo, su fuerza y su radio de acción.

Cada uno de los bandos cuenta además con una pieza clave: la bruja (para el lado oscuro) y el mago (en el lado de la luz). Estas dos piezas tienen la capacidad de invocar poderes mágicos y lanzar una serie de hechizos y encantamientos contra el adversario. Por ejemplo pueden aprisionar una pieza del bando contrario, convocar la furia de los elementos o revivir una de las piezas perdidas en alguna de las batallas.

Otro elemento que hay que tener en cuenta es el tablero de juego. Las casillas en que se encuentra dividido pueden ser blancas, negras o de color. Las blancas son favorables para el lado de la luz. Esto quiere decir que una figura del lado de la luz será más fuerte y luchará mejor si se encuentra en una casilla blanca. Las casillas negras, por su parte, son favorables a las fuerzas de la oscuridad. Por último, las casillas de color serán favorables a una u otra fuerza según el color que presenten. Este color irá cambiando de forma cíclica a lo largo del juego.

Con estos elementos, cada uno de los jugadores tiene que plantear su estrategia con el fin de ganar la partida. Para ello tendrá que, o bien colocar una de sus fichas en cada uno de los cinco puntos fuertes del tablero (casillas especiales marcadas con un círculo destellante), o bien acabar con todas y cada una de las criaturas del contrario.

Hasta ahora hemos hablado de la parte estratégica o de tablero, del juego, pero como ya hemos mencionado, hay también una parte de acción. En esta parte se entra cada vez que una pieza de uno de los bandos quiere «comerse» a una pieza del bando contrario para ocupar su casilla. Entonces la casilla en cuestión se ensancha hasta ocupar toda la



pantalla y se convierte en un campo de batalla en el que se enfrentarán a muerte las dos piezas enemigas. En este momento lo que importa es saber manejar el joystick y conseguir eliminar al contrario. Al terminar el combate, volveremos al tablero de juego y veremos cómo la pieza vencedora ocupa la casilla del combate.

Tras este argumento tan original y divertido, nos encontramos con un juego que, además, está muy bien realizado. Los gráficos y el sonido son modestos, sin grandes alardes, pero resultan adecuados a las características del juego y en nada deslucen su calidad.

La adicción está asegurada, sobre todo cuando se juega bajo la opción de dos jugadores. Ninguno de los dos tiene ventajas sobre el otro y si uno es más diestro en los combates, el otro puede intentar vencer a base de estrategia.

Otro detalle interesante es que tanto el programa como las instrucciones del mismo han sido traducidas al castellano.

FICHA DEL JUEGO

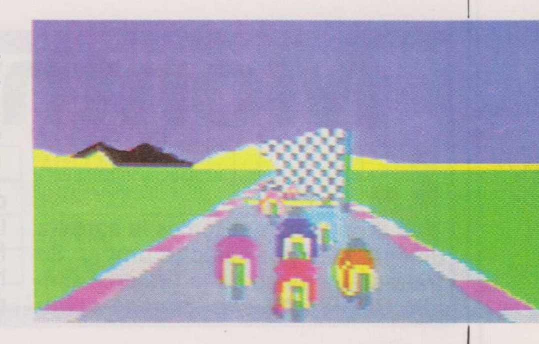
NOMBRE:	**ARCHON**
DE:	**ELECTRONIC ARTS**
ORDENADOR:	**COMMODORE-64/128**
CONTROL:	**JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD										■
ADICCION									■	
GRAFICOS					■					
SONIDO					■					
GENERAL								■		

Speed King



Atari lanzó al mercado su famoso Poole position, un juego de simulación de una carrera de fórmula uno, que pronto se convirtió en uno de los superventas del mercado. A partir de esta primera versión aparecieron otras, similares en lo fundamental, pero incorporando cada una de ellas alguna mejora o diferencia respecto del original. Este es el caso de SPEED KING. Se trata también de un



juego de simulación, pero en este caso lo que se simula es un gran premio de motociclismo. El jugador, ya sea a través del joystick o del teclado, maneja una formidable moto de competición a los mandos de la cual tratará de vencer a otros diecinueve participantes.

El manejo de la motocicleta es bastante completo y realista. Incluye la posibilidad de acelerar, frenar y cambiar de marcha, ya sea a marchas más largas o más cortas, de entre un total de seis.

Para tomar las curvas hay que tumbar la moto gradualmente hacia el interior de la curva, teniendo en cuenta que cuanto mayor sea la velocidad a la que se tomen, más fácil será salirse de la pista.

Todo ello proporciona una excelente sensación de estar conduciendo una motocicleta de verdad. Veamos cómo se desarrolla una carrera. En primer lugar nos encontramos con que tenemos que seleccionar entre una serie de opciones. La primera de ellas es la del nivel de dificultad de la carrera. Hay tres niveles distintos. Cuanto mayor es el nivel de dificultad, más deprisa conducen los adversarios, por lo que resulta más difícil adelantarles y proclamarse campeón. Una vez seleccionado el nivel de dificultad hay que elegir el circuito en el que se quiere competir. Se puede elegir entre un total de diez circuitos, réplicas exactas de los más famosos circuitos de todo el mundo (por supuesto entre ellos está nuestro circuito del Jarama). Terminada esta fase de selección se puede escoger entre comenzar la carrera o dar un vistazo rápido al circuito para hacerse una idea de cómo se ven las curvas desde el asfalto.



Comencemos la carrera. Nos encontraremos al final de una hilera de veinte motos, que echarán a andar. La nuestra sale en última posición y deberá remontar la carrera y adelantar al resto para que nos proclamemos campeones.

Aunque al principio nos quedemos retrasados, si vamos acelerando veremos acercarse a algunas de las motos que van delante nuestro. Para adelantarlas no hay más que buscar un hueco y colarse por él. Si por casualidad chocamos con alguna de las motos al intentar adelantarlas, veremos cómo nuestro piloto sale despedido y rueda por el suelo. Al cabo de unos segundos, cuando probablemente nos hayan adelantado todos los participantes, podremos reanudar la marcha.

La carrera puede desarrollarse a dos o más vueltas según elija el jugador.

Otro detalle interesante es el marcador que aparece al finalizar la carrera y en el que figuran la posición y el tiempo obtenidos por el jugador.

Todo lo que llevamos dicho define a SPEED KING como un juego de gran atractivo. Críticas apenas sí hay alguna que hacer. La definición de los sprites que constituyen cada uno de los participantes no es muy buena, le falta algo más de trabajo. También hay fallos en el movimiento de la pista, pues en determinadas ocasiones, los bordes rojos y blancos del circuito dejan de moverse y da la sensación de que estamos parados cuando en realidad vamos a cerca de 300 km. por hora.

Detalle a destacar lo constituye el sonido del motor de la moto. Responde muy bien al régimen de aceleración y a los cambios de marcha.

Resumiendo, nos encontramos ante un juego atractivo, no demasiado original y que quizá se quede un poco corto para los estándares de gráficos, sonido y movimiento a los que últimamente nos tiene acostumbrados Commodore.

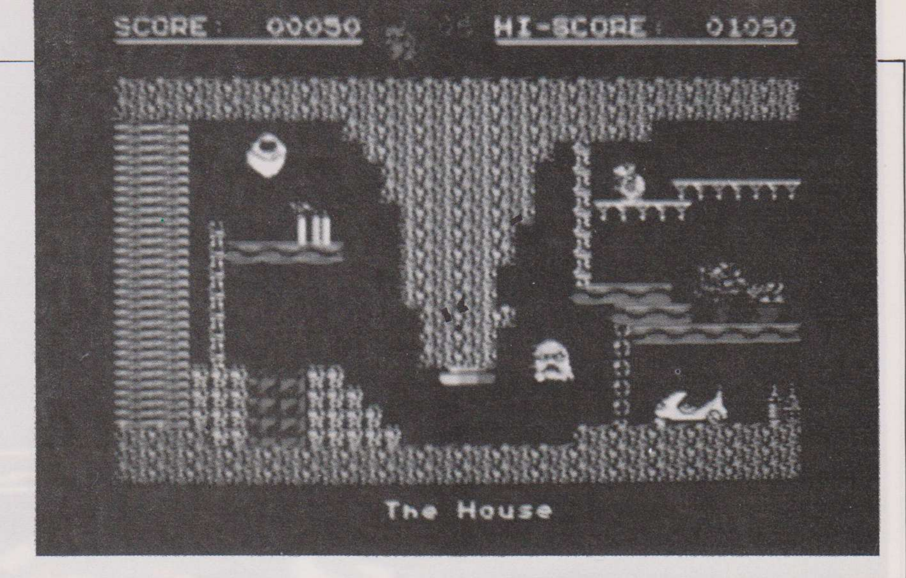
FICHA DEL JUEGO

NOMBRE:	**SPEED KING**
DE:	**DIGITAL INTEGRATION**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK Y TECLADO**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD							■			
ADICCION								■		
GRAFICOS						■				
SONIDO								■		
GENERAL								■		

Monthy on the Run

Monty on the Run es la estrella de este mes por derecho propio. Se trata de un juego de la firma GREMLIN GRAPHICS que aparece como continuación o segunda parte de otro llamado MONTY MOLE (el topo Monty). Monty on the Run narra la historia del topo Monty, que ha escapado de prisión y se encuentra en plena fuga a través de una impresionante superestructura de plataformas, pasillos, escaleras, pasadizos, trampillas y muchas cosas más. Monty se desplaza a través de la es-



tructura andando o saltando, controlado por el joystick del jugador. El objetivo final del juego es el de ayudar a Monty a encontrar un bote de remos que le permita escapar, por mar, de la justicia británica.

Monty es desde luego el mejor juego de plataformas que hemos visto hasta ahora. Y ello, en primer lugar, por su originalidad. Cada una de las habitaciones a las que se va accediendo, a medida que se avanza, está siempre llena de sorpresas, de elementos originales que hay que sortear o a través de los cuales hay que desplazarse. Por ejemplo, en algunas habitaciones hay ascensores, algunos de los cuales pueden utilizarse para subir o bajar entre niveles.

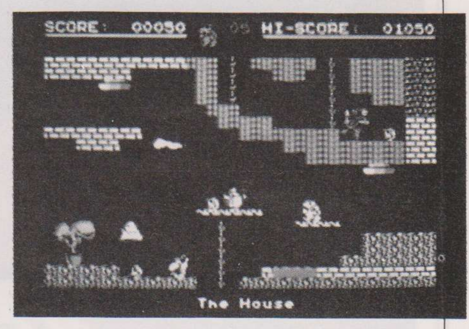
En otras pantallas aparecen los mismos ascensores, pero en este caso constituyen una trampa y destruyen una de las vidas del pequeño Monty. Para pasar de una habitación a otra puede haber cuerdas, escaleras, pasadizos y, en algunos casos, rayos teleportadores. Todo el juego es un derroche de imaginación plasmado en unos gráficos excelentes, que incluyen representaciones de escenarios de lo más barroco junto a figuras y personajes de todo tipo, como relojes voladores, cafeteras que se mueven, tazas con ojos y boca, globos, ascensores, helados, paquetes de regalo... Junto a ellos, hay también cantidad de personajes indefinibles, sprites de variadas formas y colores que van apareciendo uno tras otro de habitación en habitación. Monty debe ir recogiendo una serie de objetos que aumentarán su puntuación y le ayudarán a escapar, pero la verdad es que cuando uno se pone a jugar por primera vez, lo que menos importa es la puntuación y con lo que de verdad se disfruta es avanzando de habitación en habitación descubriendo los secretos, novedades y personajes de cada una de ellas.

Otro aspecto esencial de Monty on the Run es la música. Y decimos música porque no se trata de sonidos más o menos logrados sino de una auténtica composición musical a tres voces que suena durante todo el juego, variando, deslizándose y comenzando una y otra vez de forma casi siempre distinta.

El juego incluye tres composiciones musicales, la primera de ellas para la presentación, otra durante el juego y la última al término del mismo. Todas ellas son realmente maravillosas, lo mejor que hemos escuchado en un Commodore desde hace mucho tiempo.

Quizá no sea mucho decir que este juego merece la pena, aunque a uno no le gusten los juegos, sólo por su música. Y es que, además de las composiciones musicales que suenan durante el juego, los efectos de sonido correspondientes a ciertas situaciones son excelentes. Por ejemplo, en las primeras pantallas hay una especie de columnas que bajan desde el techo y que pueden aplastar a Monty, bueno, pues cuando esto ocurre, se oye el ruido que produce Monty al estallar y es impresionante; la primera vez que lo escuchamos pegamos un respingo en el asiento.

Otra de las características de Monty on the Run es que requiere de una gran destreza con el joystick. Al principio, las primeras veces que se juega, se tiene la impresión de que uno nunca conseguirá pasar de las dos o tres primeras pantallas (y eso que se dispone de seis vidas), así de difícil resulta. Pero cuando se adquiere un poco de práctica en el manejo del topo, la cosa cambia, y uno des-



cubre que determinadas situaciones, que parecía imposible resolver, se solucionan con un poco de habilidad. Además, normalmente, la vía de salida no es única, sino que existen varios caminos alternativos.

Al principio, mientras no se tenga mucha práctica, se puede escoger el camino fácil para, más adelante, volver atrás y probar por el camino difícil. Esto da una idea de lo difícil que va a resultar llegar a cansarse de este juego.

Con todo lo comentado sólo se puede concluir diciendo que este Monty on the Run de Gremlin Graphics (distribuido en España por SERMA S. A.) es el mejor juego de plataformas que hemos visto nunco. Las puntuaciones hablan por sí solas, hacía mucho que ningún juego había merecido en tres apartados la puntuación 10.

FICHA DEL JUEGO

NOMBRE:	**MONTY ON THE RUN**
DE:	**GREMLIN GRAPHICS**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD								■		
ADICCION										■
GRAFICOS										■
SONIDO										■
GENERAL									■	

Karateka

Cuenta la historia que tras varios años de intenso estudio con uno de los mejores maestros de kárate, el karateka regresó a su ciudad natal para encontrársela destruida, sojuzgada y dominada por las hordas del tirano Akuma. Por si fuera poco, la familia y amigos del karateka habían desaparecido. Además, su amada, la preciosa princesa Mariko se encontraba prisionera en la fortaleza del tirano. El valeroso karateka tomó una decisión: armado tan sólo con sus conocimientos de kárate se enfrentaría a los esbirros de Akuma, rescataría a su princesa y acabaría de una vez por todas con el malvado Akuma.

Esta es la historia en la que se basa el juego del karateka, un juego de intensa acción en el que, en unos escenarios gráficos de impresionante realismo, la figura del karateka deberá enfrentarse sucesivamente a los enemigos que se interpongan en su camino.



El juego, del que hay versiones en cassette o en diskette, comienza con una muy lograda presentación en la que podemos ver cómo la princesa Mariko, encerrada por Akuma en una habitación de su fortaleza, llora amargamente su desdicha. Ya en esta presentación nos hacemos una idea de la calidad gráfica del juego y de lo cuidado de los movimientos de los sprites, que más que sprites parecen dibujos animados.

Después de la presentación hay que volver a cargar un segundo trozo de programa para empezar a jugar. Esta es una de las importantes pegas que hemos encontrado en este programa, pues no sólo hay que detener-



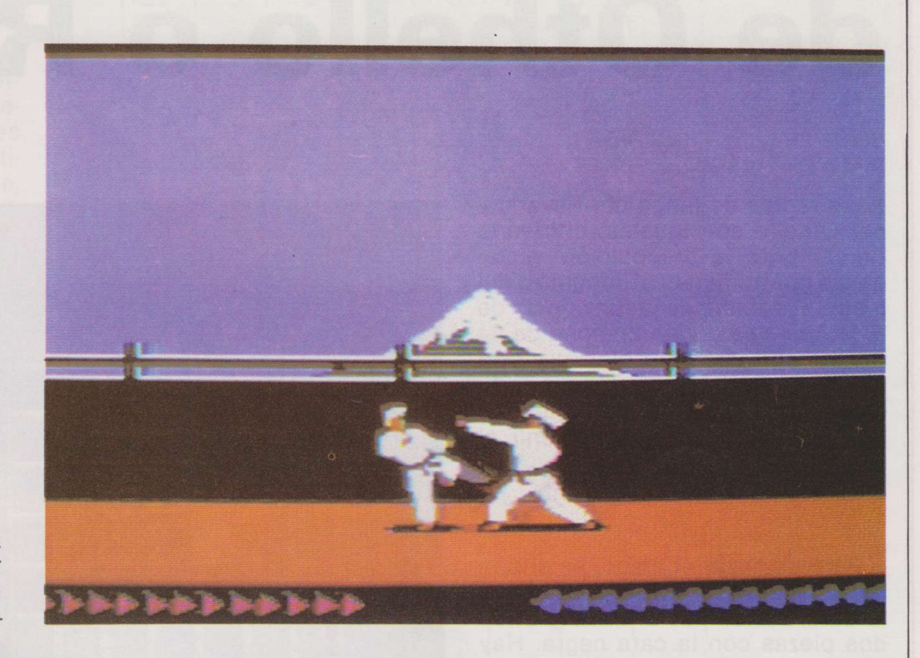
se a cargar trozos cada vez que se supera una de las fases del juego, sino que además, cada vez que el karateka pierde la única vida que posee, hay que comenzar a cargar el programa de nuevo. Este proceso ya resulta lento y tedioso cuando se utiliza la unidad de diskettes, así que imaginemos lo que ocurre cuando se utiliza el cassette...

Una vez que el juego ha comenzado, nos encontramos controlando los movimientos del karateka, al que podemos ver en primer plano sobre la pantalla. Se puede utilizar tanto el joystick como el teclado pero, curiosamente, el manejo resulta más cómodo y más preciso desde el teclado (algo poco corriente en la mayoría de los juegos). Pulsando las teclas adecuadas es posible hacer que el karateka corra, se detenga y se prepare para el combate. Una vez que está preparado para combatir, podrá lanzar golpes dirigidos a la cabeza, al tronco o a las piernas de su adversario, y esto tanto con el puño como con los pies, lo que nos da un total de seis golpes distintos.

Con una combinación adecuada de estos seis golpes hay que vencer a los sucesivos enemigos que irán apareciendo, uno tras otro, por el lado izquierdo de la pantalla. La estrategia en el combate es importante. Por una parte hay que propinar cuantos golpes sea posible en el menor tiempo de forma que vaya agotándose la energía del contrario. Esta energía aparece representada en la parte inferior de la pantalla mediante una serie de flechas. Cuando consigamos eliminar todas las flechas, el contrario se desplomará y habremos vencido el combate. Pero al mismo tiempo, hay que tener cuidado de no exponer al karateka ya que cuantos menos golpes reciba, más tiempo conservará su energía. Cuando el nivel de la misma sea muy bajo, es conveniente dedicarse a esquivar los golpes, con lo que el nivel energético se recuperará.

A medida que se vence a cada uno de los esbirros de Akuma, hay que lanzar al karateka hacia adelante y avanzar todo lo posible antes de que aparezca el siguiente. Por regla general, cada contrario es un poco más difícil de vencer que el anterior.

El juego se desarrolla así, de esta manera, en una sucesión de combates de un extraordinario realismo, sobre todo en lo que se refiere al movimiento de los sprites. Es una lástima que el movimiento no sea un poco más rápido y que el karateka no disfrute más que de una vida, perdida la cual hay que recomenzar con el tedioso proceso de carga del programa.



Salvo estas pequeñas pegas el juego resulta, en nuestra opinión, bastante recomendable.

FICHA DEL JUEGO

NOMBRE: **KARATEKA**
DE: **BRODERBUND**
ORDENADOR: **COMMODORE-64/128**
CONTROL: **JOYSTICK/TECLADO**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD							■			
ADICCION							■			
GRAFICOS										■
SONIDO					■					
GENERAL						■				

Renaissance: una versión de Othello o Reversi

Las reglas de juego del Reversi y del Othello son iguales, difiriendo únicamente la disposición inicial de las cuatro fichas al comienzo del juego. El tablero de juego consta de 8 × 8 casillas, y se dispone de 64 fichas, con una cara blanca y otra negra. Es un juego en el que participan dos jugadores, con igualdad de oportunidad al no intervenir el azar.

Para empezar el juego, se sitúan en el cuadrado central 2 piezas con la cara blanca hacia arriba y otras dos piezas con la cara negra. Hay dos variantes:

— en el Othello, las fichas del mismo color estan en diagonal, y

— en el Reversi, las piezas del mismo color están en la misma fila.

Comienza a mover el jugador que conduce las fichas negras. Alternativamente van jugando los dos contendientes, cediendo el turno aquel jugador que no pueda situar una nueva ficha en una posición legal. El juego concluye cuando se han situado sobre el tablero todas las piezas o cuando ninguno de los dos jugadores puede poner ninguna ficha de acuerdo con las normas de captura.

Cada nueva pieza que se sitúa en el tablero debe encerrar, junto con alguna otra ficha del mismo color ya presente, a alguna o varias piezas enemigas, que quedan capturadas, es decir, que cambien de color, lo que se realiza invirtiéndolas (de ahí el nombre de reversi). Las fichas contrarias pueden ser capturadas en horizonta, vertical o diagonal.

El juego es muy dinámico porque la aparente ventaja material se modifica rápidamente, sobre todo en las últimas jugadas. Hay casillas cuya ocupación es definitiva: las cuatro esquinas. El valor estratégico de las casillas se descubre en la práctica tras jugar muy pocas partidas.

Este juego de tablerro esta bastante difundido, aunque sin haber alcanzado las cotas del ajedrez, las damas o el chaquete (backgammon). Su nivel de complejidad es semejante al de las damas, e inferior al de ajedrez.

Cada día hay más adeptos a este juego, pero la estrecha correspondencia entre la calidad del juego y el número de partidas jugadas, puede restar atractivo para los jugado-

res más bisoños, que son contundentemente derrotados por jugadores con mayor experiencia. Por esta razón, contando con un programa como Renaissance se puede mejorar en poco tiempo el nivel de maestría y dar un susto a algún amigo experto en Reversi (—lo digo por experiencia—).

MAQUINAS Y PROGRAMAS DEDICADOS AL OTHELLO

Existen varias versiones del Othello para microordenadores. Fidelity Electronica, uno de los mayores fabricantes de máquinas especializadas de ajedrez, damas, chaquete, brigde, etc, ofrece un modelo dedicado en exclusiva al Reversi, la máquina Reversi Sensory Challender. Dispone de un tablero sensible que percibe y registra automáticamente los movimientos. Cuenta varios niveles de destreza, modalidad de resolución de problemas, modalidad de supervisor para juego de dos humanos, cambio de lado, permite el retroceso de varias jugadas, sugiere un contrajugada si se solicita, etc.

Victor Castro, un lector de nuestra revista envió un programa Othello 64 que mereció ser premiado y publicado en el número 12 de COMMODORE MAGAZINE. Este programa de Othello, que juega razonablemente bien, puede ser transformado en el juego del Reversi, sin más que modificar la línea 730. En la secuencia de números aparecen 1, que deben cambiarse a -1, 1, resultando la nueva línea 730:

```
730 DATA 6,2,6,-1,1,6,2,6,0,0,
6,2,6,-1,1,6,2,6,0,0,
8,2,8,6,6,8,2,8,0,0.
```

También hemos podido jugar también con programas de Othello escritos para otros ordenadores.

Primera Partida

Tiempo total: 22 minutos.

Negras: Renaissance. Final con 21 fichas.
Blancas: El autor. Final con 42 fichas.

--	36	35	34	24	25	26	58
30	52	37	33	31	23	57-1	44
13	28	2	1	4	7	8	27
29	9	3	0	0	12	17	43
10	16	6	0	0	32	39	40
19	22	11	14	46	5	45	41
20	55	50	47	15	38	53	42
56	57	51	21	48	18	49	54

Segunda Partida

Tiempo total: 23 minutos.

Negras: El autor. Final con 36 fichas.
Blancas: Renaissance. Final con 28 fichas.

49	48	39	49-1	51	50	51-1	47
37	--	38	51-2	44	--	46	29
36	33	30	25	32	17	31	28
35	34	26	0	0	16	27	21
41	40	24	0	0	12	20	22
42	45	3	2	1	4	5	8
43	--	15	6	9	13	--	23
--	19	14	18	7	10	11	--

RENAISSANCE

Se trata de un programa escrito en 1982 por Louis Savain, presentado por Audiogenic de Gran Bretaña. Se vende en cassette al aquilatado precio de menos de 3.000 pts. Trás casi tres minutos de carta, aparece una carta de presentación y luego se muestra la pantalla de juego, con el tablero y un menú de opciones. A modo de demostración se está desarrollando una partida dirigiendo el programa ambos bandos. Se pide el nombre del jugador humano, para personalizar el juego y el color de las piezas que quiere jugar.

Puede jugarse mediante teclado, aunque se ofrece la oportunidad de

simplificar la entrada de movimientos mediante joystick.

Las funciones opcionales admitidas son:

PL de Play, para que el programa juegue.

CH de Change colour, para cambiar de lado, pasando a dirigir las fichas del contrario.

TA de Takes back moves, para retroceder la partida hasta el movimiento que se desee.

SE de Set board, para establecer una posición de fichas que se quiera analizar, permitiendo resolver problemas de este juego.

HE de Help, para pedir que el programa nos sugiera una buena jugada.

LE de Level, para definir el nivel de juego del programa, entre 1 y 8.

Renaissance muestra en pantalla el número de jugada actual, las últimas posiciones ocupadas por cada jugador, el número de fichas de cada color y el nivel de juego, así como una indicación de a quien le corresponde jugar. Cuando esta "pensando" su jugada parpadea el indicador PL y el mensaje MY TURN. El análisis finaliza cuando el usuario pulsa RETURN o el botón de disparo del joystick. Para efectuar una jugada basta llevar con las teclas de cursos o mediante el jostick una señal hasta la casilla deseada y pulsar RETURN o el botón de disparo. El programa rechaza las jugadas que incumplen el reglamento del juego, pudiendo incluso recomendar un buen movimiento, excepto si el nivel elegido es el inferior. Se hubiese agradecido que Renaissance, al igual que otros programas con los que hemos podido jugar, indicase todas las posibles casillas que pueden ser ocupadas en la próxima jugada del humano, designándolas con letras para simplificar la elección.

Una enervante, además de totalmente innecesaria, característica que han incorporado a este programa es la aparición de un círculo rojo en las distintas casilla que pueden ser alcanzadas en la siguiente jugada, según las previsiones efectuadas. Esta rápida y reiterativa aparición del circulo, que no aclara prácticamente nada de la evolución de la partida, obliga a jugar solamente en los niveles inferiores para no prolongar desmedidamente el tiempo ocupado pór el programa. Por ejemplo, en el nivel 5 del juego medio de una partida tras reflexionar la máquina 30 segundos, puede dedicar más de 8 minutos a esta secuencia inteminable de círculos rojos revoloteando de una casilla a otra. Este defecto desmerece en gran medida un programa, que por lo demás, tiene suficiente calidad y gama de opciones. Personalmente, nos gusta que opcionalmente el usuario pueda "presenciar como razona" el programa, pero sin interferir el modo habitual de juego que es el más utilizado.

OTHELLO — POSICION INICIAL — REVERSI

CALIDAD DE JUEGO

Renaissance tiene un nivel de juego superior al de los otros proquina especializada que hemos mencionado, pero esta calidad de juego no es aprovechable, salvo que se destine toda una tarde para jugar una partida, por la desafortunada característica que hemos comentado. Para un jugador novel de Othello es más ameno jugar con el programa enviado por el lector, que ademá es gratis, que enfrentarse al nivel 2 de Renaissance.

El jugador aficionado, que busca un adversario más cualificado o que desea mejorar su juego, debe recurrir a Renaissance en los niveles del 3 al 5, si quiere un poco de agilidad, y tomándoselo con mucha paciencia, puede jugar en alguna ocasión con los niveles superiores.

El algoritmo de búsqueda de Renaissance es especialmente eficiente en los finales de juego, causando en ocasiones auténticos quebraderos de cabeza al oponentes.

Aprovechen los buenos aficionados esta epoca histórica, irrepetible en la que todavía pueden ganar al Othello o al Ajedrez a cualquier programa de microordenador que exista. Dentro de muy poco esta carrera entre inteligencias, humana versus artificial, tendra un resultado definitivo.

Dado que la calidad se aprecia reproduciendo algunos juegos, hemos recogido dos partidas perdidas por Renaissance en el nivel 3, jugando con negras y con blancas. Estas Partidas pueden ser comprobadas, ya que el programa no emplea ninguna rutina aleatoria, sino que juega, en el mismo nivel y configuración, el mismo movimiento.

La notación seguida indica el movimiento en el que se ocupó cada escaque, siendo impares las jugadas del negro y pares los movimientos del blanco. Cuando algún jugador repite movimientos porque el contrario no tiene ninguna casilla disponible, se indica con un tercer número (por ejemplo, 57-1 significa que tras el movimientos 57, el negro repite jugada). □

Una creación de Electronic Arts producida y presentada por DRO SOFT con instrucciones y pantallas en CASTELLANO.

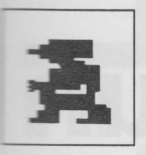
OGRO



DRAGON



TROLL

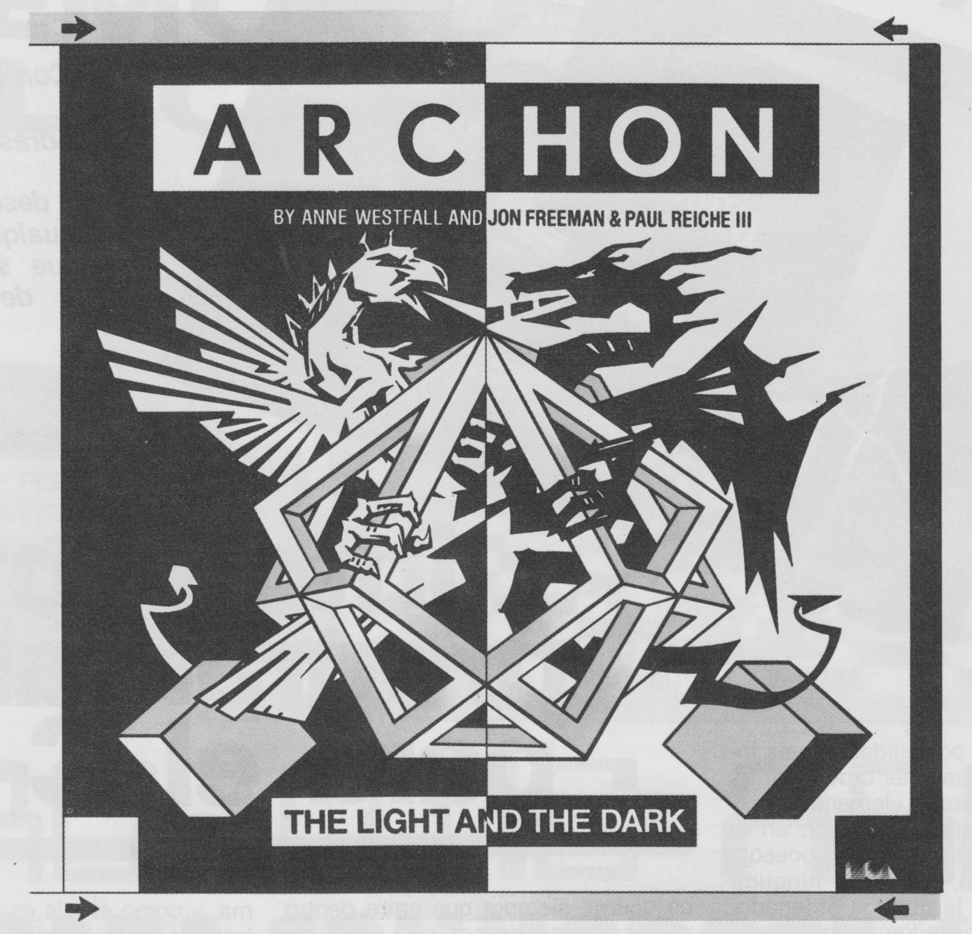


BASILISCO



BRUJA





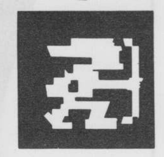
FENIX



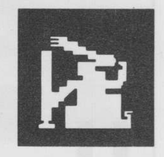
UNICORNIO



ARQUERO



MAGO



CABALLERO



En la época actual, estamos viviendo el resurgimiento del género de la fantasía y la aventura, dentro de esta corriente, JON FREEMAN, el principal creador de este juego elaboró un ajedrez fantástico cuyos personajes podían estar sacados de cualquier aventura de mazmorras y dragones (DUNGEONS AND DRAGONS).

ARCHON.

ARCHON presenta la eterna lucha entre la luz y la oscuridad, en un contexto de acción y estrategia entre fuerzas opuestas sacadas del mito y la leyenda. El lado de la luz y el lado oscuro son iguales en números y equivalentes en fuerza, pero no idénticos, su objetivo sin embargo es el mismo, conquistar los cinco puntos de poder o eliminar a la facción opuesta.

Las diferentes criaturas fantásticas bajo control de los jugadores forman 18 por bando de 8 clases diferentes.

Ninguna de las suyas son iguales a las de su oponente.

Archon se juega en dos pantallas. En una (la estratégica) los jugadores alternan turno moviendo sus piezas a posiciones favorables. Cuando se coloca una pieza en un cuadro ocupado por una pieza enemiga, el juego pasa temporalmente a la segunda presentación, la arena de combate. En un clásico juego de tablero, el defensor es barrido de su casilla sin mayor discusión. En Archon no se admiten tranquilidades de ese tipo; si alguien quiere algo tiene que luchar por ello. Cuando intente entrar en un cuadro ocupado por el enemigo lo verá: el cuadro en disputa se expande hasta llenar la pantalla... entonces en palabras de cierto monstruo de piel naranja del comic ¡ES LA HORA DE LOS MAMPORROS!



ARCHON



DRO SOFT

DRO SOFT. Fundadores, 3. Madrid 28028



ELECTRONIC ARTS

Montaje



Con este sencillo montaje, podrás controlar la conexión y desconexión de cualquier aparato que se te ocurra, desde tu C-64.

un interruptor programable para el C-64.

Una de las posibilidades más interesantes de cualquier ordenador es la de utilizarlo como elemento de relación con el mundo exterior, en el control de determinados procesos. Quizá sea esta, más que ninguna otra, la vía por la que los ordenadores y microprocesadores van a revolucionar nuestra sociedad. De hecho, actualmente, las industrias estan introduciendo sistemas de control automático de procesos por ordenador en muchos procesos de producción, con el consiguiente incremento de productividad que ello supone.

Pero no sólo en la industria, sino también en muchas situaciones de la vida cotidiana ha irrumpido con fuerza *el microordenador como controlador de procesos*, por ejemplo en algunos autómoviles y en ciertos electrodomésticos. *A nadie le extraña* ya la idea de un futuro en el que gran parte de los sistemas de una ciudad y de los hogares de dicha ciudad funcionen de forma automática bajo el control de pequeños «chips» microprocesadores. Con esta idea en la cabeza, aunque con unas pretensiones por el momento mucho más modestas hemos preparado para vosotros este montaje. Consiste en la conexión de un sencillo RELE a vuestro C-64 que podréis utilizar a modo de interruptor programable y con el que podreis controlar la conexión y desconexión de cualquier aparato que se os ocurra, siempre que entre dentro de los límites de potencia para los que ha sido diseñado el montaje. Por ejemplo, y sólo por citar una de las innumerables aplicaciones de este montaje, podemos imaginar un hipotético hogar en el que a las siete y media de la mañana un ordenador Commodore pone en funcionamiento una cafetera. Cinco o diez minutos más tarde, cuando el café está preparado, el Commodore se encarga de *desconectar la cafetera y se dedica a* la ingrata tarea de despertar a su «señor» mediante una serie de pitidos agudos repetidos a intervalos de un minuto. Esta y cualquier otra aplicación que se os ocurra la podréis llevar a cabo si leéis detenidamente este artículo y seguis con cuidado las indicaciones que en él mismo os vamos a dar.

CARACTERISTICAS DEL MONTAJE

Como ya hemos dicho, el montaje consiste en la conexión de un relé al port de usuario del ordenador. De este modo, como el port de usuario se puede controlar desde un programa, y como el relé es capaz de controlar la conexión de un aparato a la red de energía eléctrica, se podrá controlar cualquier aparato desde un programa. El diseño ha sido concebido para que sea posible conmutar corriente alterna de la red de 220 voltios o 125 voltios. De todas formas también puede conmutar corriente continua. En cualquier caso y con el relé que hemos elegido, el máximo valor de corriente que se puede conmutar *es de tres amperios (3A)* tanto en alterna como en continua. Como la potencia de un aparato eléctrico viene dada por la sencilla fórmula...

$$P = V \times I$$

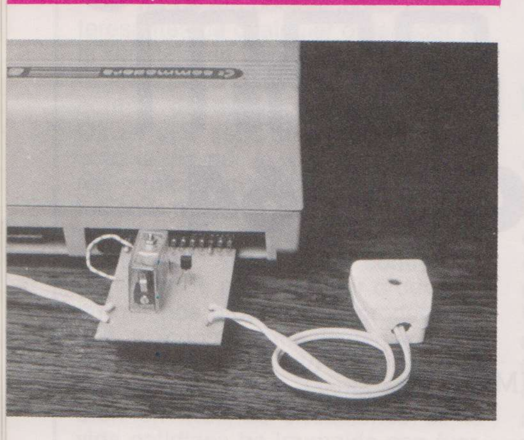
... es decir, la potencia es igual al producto de la tensión por la intensidad de la corriente, con una tensión de 220 voltios y una densidad máxima de 3 amperios podremos controlar aparatos con una potencia máxima de unos 660 vatios. Si trabajamos con una tensión de 125 voltios la potencia máxima de trabajo se reduce a 375 vatios. Este cálculo nos dice que el montaje podrá controlar hasta 6 bombillas de 100 vatios, o cualquier televisor o un aparato de música o un

pequeño calentador... y en general, cualquier aparato que os pueda ocurrir.

Además, y para aquellos a quienes les parezca poca la potencia que se pueda controlar, diremos que esta potencia (más concretamente la corriente) sólo depende de los contactos del relé, es decir que con un relé que soporte una corriente de 10 amperios, se podrá, con el mismo circuito, controlar una corriente de 10 amperios o lo que es lo mismo una potencia de unos 2200 vatios.

FUNCIONAMIENTO

El funcionamiento del circuito es muy sencillo y vamos a explicarlo teniendo en cuenta el esquema del mismo que es el que aparece en la figura 1. Pero antes vamos a hablar un poco del funcionamiento del port de

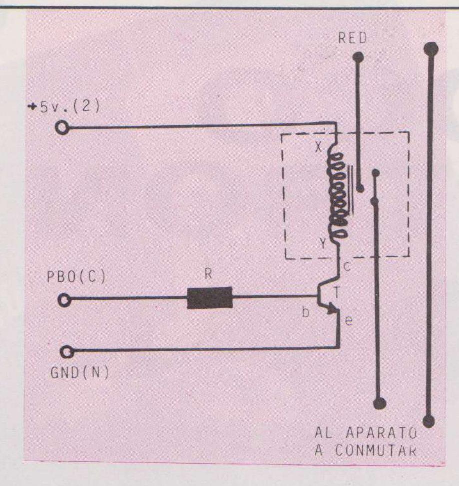


usuario. El port de usuario es un conector de la parte trasera del ordenador que internamente va conectado a uno de los «chips» del mismo. El port de usuario tiene muchas salidas, pero para nuestro montaje sólo vamos a utilizar tres de ellas que son...

TERMINAL C (PBO)
TERMINAL N (GND)
TERMINAL 2 (5v)

Los tres terminales pueden verse en la figura 2.

El primero de ellos, el terminal C (PBO) es el que permite controlar el circuito desde un programa. Como veremos más adelante, existe una posición de memoria en la que uno de

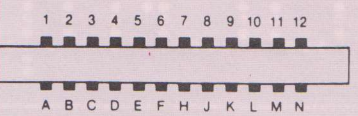


los bits se corresponde con este terminal. De este modo cuando el bit se ponga a «1», mediante un POKE o por cualquier otro procedimiento, el terminal C (PBO) tomará una tensión de 5 voltios. Del mismo modo, cuando dicho bit se ponga a «0», el terminal reducirá su tensión a 0 voltios. En esto se basa el funcionamiento del montaje. Cuando el terminal se pone a 5 voltios, el relé se excita y se cierran sus contactos. Cuando el terminal se pone a 0 voltios los contactos se abren. El circuito incluye además una resistencia y un transistor que son los señalados como R y T. La misión de estos dos componentes es la de proporcionar la corriente necesaria para el relé sin sobrecargar la

User I/O

Pin	Type	Note
1	GND	
2	+5V	MAX. 100 mA
3	RESET	
4	CNT1	
5	SP1	
6	CNT2	
7	SP2	
8	PC2	
9	SER. ATN IN	
10	9 VAC	MAX. 100 mA
11	9 VAC	MAX. 100 mA
12	GND	

Pin	Type	Note
A	GND	
B	FLAG2	
C	PB0	
D	PB1	
E	PB2	
F	PB3	
H	PB4	
J	PB5	
K	PB6	
L	PB7	
M	PA2	
N	GND	



salida del terminal C (PBO). Los otros dos terminales del port que se utilizan son el terminal 2 y el terminal N que proporcionan respectivamente cinco voltios y masa (o lo que es lo mismo cero voltios). En la figura 3. hemos representado cuales son las patillas o terminales del transistor y cuales los del relé. Por último en la figura 4. aparece un posible diseño para una placa de circuito impreso en la que se acoplen todos los componentes. En cualquier caso el circuito se puede montar tanto en la placa de circuito impreso como en cualquier otra placa para montajes de las que se pueden encontrar en el mercado.

REALIZACIÓN DEL MONTAJE Y CONEXION AL ORDENADOR

La realización del montaje es muy sencilla, no hay más que seguir el esquema de la figura 1. El único punto con el que habrá que tener un poco de cuidado es el de la correcta conexión de los terminales del relé y del transmisor. Para ello no hay más que fijarse en la fifura 3. en la que vienen representadas las patillas o terminales de estos dos elementos. En cuanto a la adquisición de los componentes no tiene que haber ningún problema. Basta con acercarse a cualquier tienda de componentes electrónicos con la lista de componentes que hemos incluido y pedir que nos los sirvan.

RECOMENDACIONES

— El montaje se puede hacer en placa impresa o en placa para montajes.

— Al soldar el transistor hay que tener cuidado para no calentarlo demasiado, ya que se podrá dañar.

— Antes de conectar ningún aparato a la red se puede probar el circuito escuchando y viendo si se cierran y abren los contactos del relé. Cuando se haya comprobado que el funcionamiento es correcto entonces se podrá conectar lo que se quiera.

— Cuidado para no confundirse con los terminales del transistor ni con los del relé.

— El montaje no puede dañar al Commodore, pero siempre es

TRANSISTOR(T)

C B E

conveniente tener cuidado al enchufar a la red ya que nos podríamos dañar nosotros mismos.

LISTA DE COMPONENTES

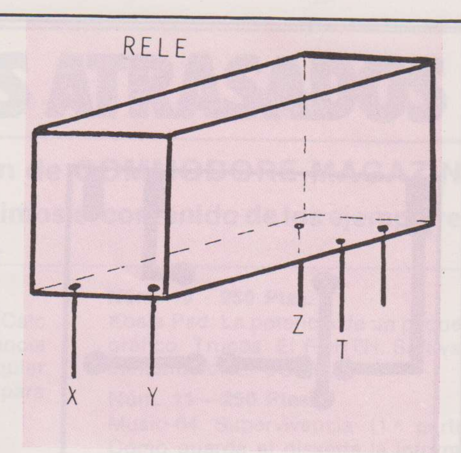
Al acercarse a cualquier tienda de electrónica a comprar los componentes para el circuito hay que pedir lo siguiente:

R. Una resistencia de 4K7 (o lo que es lo mismo, de cuatro mil setecientos ohmios) y de 1/4 de vatio (que es la potencia que puede disipar). Si no hubiera más que resistencias de 1/2 vatio daría lo mismo, valen igual.

T. Un transistor BC 548 A. Este es el transistor que hemos utilizado en el circuito, pero cualquier transistor de similares características puede valer igualmente.

RELE. El relé que hemos utilizado es un ASETYC SS1306 con una resistencia interna de unos 80 ohmios. En este caso como en el anterior, se podrá utilizar cualquier otro relé de caracteristicas similares. Sólo es necesario que pueda conmutar hasta 3 amperios. En la tienda nos podrán aconsejar. Lo que sí es importante es no confundirse con los terminales del relé ni con los del transistor. Para ello hemos incluido la figura 3.

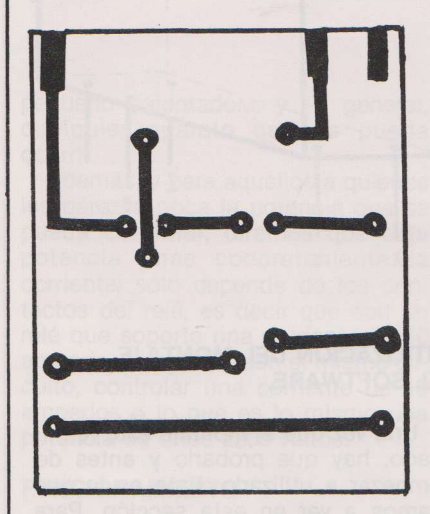
CONECTORES. Los conectores que van a hacer falta son: uno para el port de usuario del Commodore 64 y otros dos, macho y hembra, para la conexión a la red y al aparato que queramos controlar.



UTILIZACION DEL MONTAJE EL SOFTWARE.

Una vez que el montaje esté terminado, hay que probarlo y antes de empezar a utilizarlo. Esto es lo que vamos a ver en esta sección. Para ello vamos a volver al port de usuario. Dijimos que de este port utilizabamos tres terminales. Dos de ellos, los terminales 2 y N, corresponden a la alimentación del circuito. El terminal 2 es como si dijéramos el polo positivo, mientras que el terminal N corresponde al polo negativo.

El totro terminal, el terminal C, es el que realmente nos va a permitir el



control desde programa. Este terminal está conectado internamente a un terminal de un «chip» (que se denomina CIA 1). Este «chip» proporciona ocho terminales que pueden utilizarse como entradas o salidas de datos. Estos ocho terminales constituyen un port. Los terminales se suelen denominar con los nombres de PBO, PB1...PB7, de los que nosotros, como hemos dicho, sólo utilizaremos uno, el PBO que es el que va conectado al terminal C.

Pues bien, hay dos posiciones o direcciones de memoria que permiten controlar el estado de cada uno de los ocho terminales. Una de ellas, que llamaremos DDR sirve para que el Commodore sepa si el terminal es una entrada o una salida. Cada uno de los ocho terminales PBO...PB7, se corresponde con uno de los ocho bits de la posición DDR. Si el bit que corresponde al terminal está a uno «1» entonces el Commodore sabe que el terminal es una salida, mientras que si el bit está a cero «0», se trata de una entrada. En nuestro montaje, nos interesa que el bit esté a uno, ya que vamos a utilizar el terminal como salida. Así pues, en cualquier programa conel que queramos controlar el relé habrá que incluir la instrucción POKE DDR,255, con la que pondremos a uno todos los bits de DDR y definiremos los terminales del port como salidas.

Una vez que hayamos hecho esto, nos centraremos en la dirección memoria que llamaremos PORT. Entonces, para activar el relé no tendremos más que poner a uno el bit 0 de dicha dirección, lo que podremos hacer con un POKE PORT,1. Para desactivar el relé bastará con poner a cero el bit, lo que se conseguirá con un POKE PORT,0. Las direcciones correspondientes a DDR y a PORT son las tablas 1. y son las que habrá que incluir en el programa en lugar de las palabras DDR y PORT.

SERVICIO DE EJEMPLARES ATRASADOS



Complete su colección de COMMODORE MAGAZINE

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 1 - 250 Ptas.
Análisis de la nueva serie 700/ Calc result a fin de cuentas/Más potencia con Victree/Cómo adaptar cualquier cassette/Juegos y aplicaciones para VIC-20 y CBM 64.

Núm. 2 - 250 Ptas.
CBM 64 en profundidad/Superbase 64: el ordenador que archiva/Juegos, trucos y aplicaciones.

Núm. 3 - 250 Ptas.
Magic Desk, el despacho en casa/Herramientas para el programado/Interfaces para todos.

Núm. 4 - 250 Ptas.
El 64 transportable revisado a fondo/ Interface RS 232 para el VIC-20/Juegos/El fútbol-silla en su salón.

Núm. 5 - 250 Ptas.
Programas, juegos y concurso/Londres: Quinta feria Commodore/Basic, versión 4.75.

Núm. 6 - 250 Ptas.
El misterio del Basic/Lápices ópticos para todos/Concurso, juegos, aplicaciones.

Núm. 7 - 250 Ptas.
El ordenador virtuoso. MusiCalc. Programa monitor para el 64. Lápices ópticos. Ampliación de memoria para Vic-20.

Núm. 8 - 250 Ptas.
Joystick y Paddle para todos. Misterio del BASIC. EL LOGO. Cálculo financiero. Programas.

Núm. 9 - 250 Ptas.
Conversión de programas del Vic-20 al C-64. Móntate un paddle. Identifica tus errores. Software comentado.

Núm. 10 - 250 Ptas.
Koala Pad: La potencia de un paquete gráfico. Trucos. El FORTH. Software comentado. El LOGO.

Núm. 11 - 250 Ptas.
Music-64 Supervivencia (1.ª parte). Cómo guarda el diskette la información. Sintetizador-64. El Forth (1.ª parte).

Núm. 12 - 250 Ptas.
Commodore-16 por dentro y por fuera. Sprites: los alegres duendecillos (1.ª parte). Supervivencia (1.ª parte). El Forth (y 3.ª parte).

Núm. 13 - 250 Ptas.
Análisis: programas de ajedrez. Los Cazafantasmas, 64. Vic en el espacio. La impresora que dibuja. Interface paralelo.

Núm. 14 - 250 Ptas.
Sprites; cómo entenderse con los duendes. Pilot: un lenguaje de alto nivel. Guía de Software para C-64.

Núm. 15 - 250 Ptas.
Síntesis de voz: su ordenador tiene la palabra. Pilot: un lenguaje de alto nivel (2.ª parte). Guía de software para C-64 (2.ª parte).

Núm. 16 - 250 Ptas.
Análisis de simuladores: vuela con tu C-64. Contabilidad para pequeños negocios. Como acelerar la ejecución de gráficos en BASIC. Submarino Commander. Pilot: un lenguaje de alto nivel (3.ª Parte).

Corte y envíe este cupón a: COMMODORE MAGAZINE
Bravo Murillo, 377 - Tel. 7337969 - 28020-MADRID

SERVICIO DE EJEMPLARES ATRASADOS

Ruego me envíen los siguientes ejemplares atrasados de COMMODORE MAGAZINE: ____________

El importe lo abonaré:
Contra reembolso ☐ Adjunto Cheque ☐ Con mi tarjeta de crédito ☐
American Express ☐ Visa ☐ Interbank ☐ Fecha de caducidad: ______

Número de mi tarjeta: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

NOMBRE ____________

DIRECCION ____________

CIUDAD ____________ D.P. ______

PROVINCIA ____________

Cursillo sobre código máquina (III)

Una vez vistas las bases del sistema binario, y la forma básica de trabajar de nuestro ordenador, vamos (a comenzar) a aprender a programar en código máquina. Para ello vamos a centrar este capítulo en el estudio de una herramienta básica para todo tipo de programación, tanto en código máquina como en alto nivel: los organigramas.

La forma más común de programar es sentarse delante del ordenador con una idea más o menos vaga de lo que queremos hacer y empezar a programar «sobre la marcha». Esto, aunque siempre incorrecto, puede producir resultados aceptables cuando programamos en alto nivel (y siempre suponiendo un programador experto y una gran dosis de suerte).

Programar en lenguaje máquina implica, sin embargo, el tener que bajar a un nivel tan detallado en la solución del problema, que la práctica anterior suele dar resultados desastrosos, en cuanto a tiempo empleado y resultados conseguidos.

Por tanto, SIEMPRE, pero más aún cuando estamos programando en código máquina, es necesario, antes de sentarse a programar, emplear el tiempo que sea preciso en definir exactamente el problema a resolver, los resultados a los que queremos llegar y los pasos a seguir para lograrlo. El resultado de todos los análisis anteriores se plasmará en el organigrama del proceso.

Con un organigrama correctamente realizado, el pasar a la codificación del programa es inmediato, y el tiempo de depuración, hasta que el programa funciona correctamente, se reduce considerablemente.

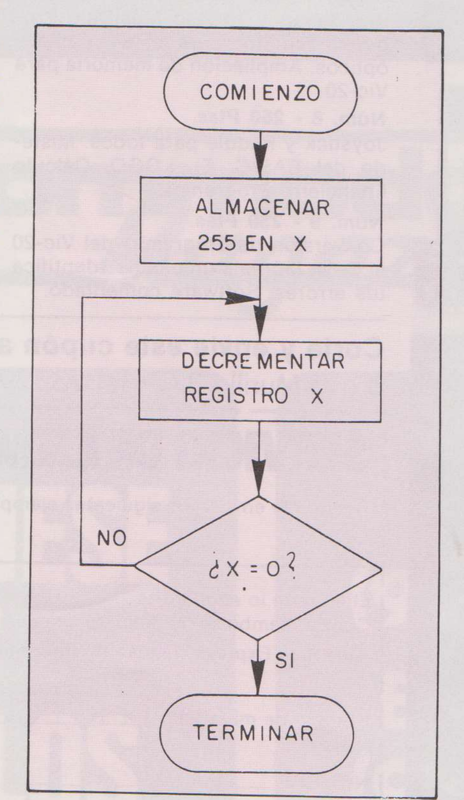


Fig. 1.

¿QUE ES UN ORGANIGRAMA?

En un sentido amplio, un organigrama no es más que un diagrama que nos muestra un proceso o programa como una secuencia de acciones y decisiones elementales (al fin y al cabo un programa no es más que eso).

Luego, el organigrama muestra cómo se suceden lógicamente esas acciones y decisiones elementales a lo largo del desarrollo del programa.

Como ejemplo, vamos a fijarnos en el organigrama de la figura 1. Este organigrama corresponde a una rutina típica de retardo.

A la vista de la figura anterior nos damos cuenta rápidamente de lo que hace nuestro programa. Unicamente almacena un valor inicial (en este caso 255) en uno de los registros de la CPU 6510, el registro X, y lo decrementa una vez en cada pasada hasta que se hace cero. El tiempo que tarde el ordenador en realizar ese proceso será el retardo producido. Vemos además que ese retardo puede variarse en función del número que introduzcamos inicialmente en el registro X.

ELEMENTOS DE UN ORGANIGRAMA

En el organigrama de la figura 1 encontramos además los principales

COMIENZO

¿SABEMOS NUMERO?

NO

BUSCAR NUMERO

SI

¿TELEFONO COLGADO?

NO

COLGARLO

SI

DESCOLGAR TELEFONO

¿RECIBIMOS TONO?

NO

ESPERAR UN TIEMPO MAXIMO

¿TONO RECIBIDO?

TELEFONO AVERIADO

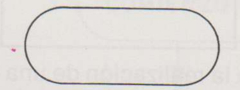
FIN

MARCAR NUMERO

FIN

Fig. 2.

bloques que van a constituir todos los organigramas. Dichos bloques son los que mostramos a continuación:



: Indica comienzo y final de programa.

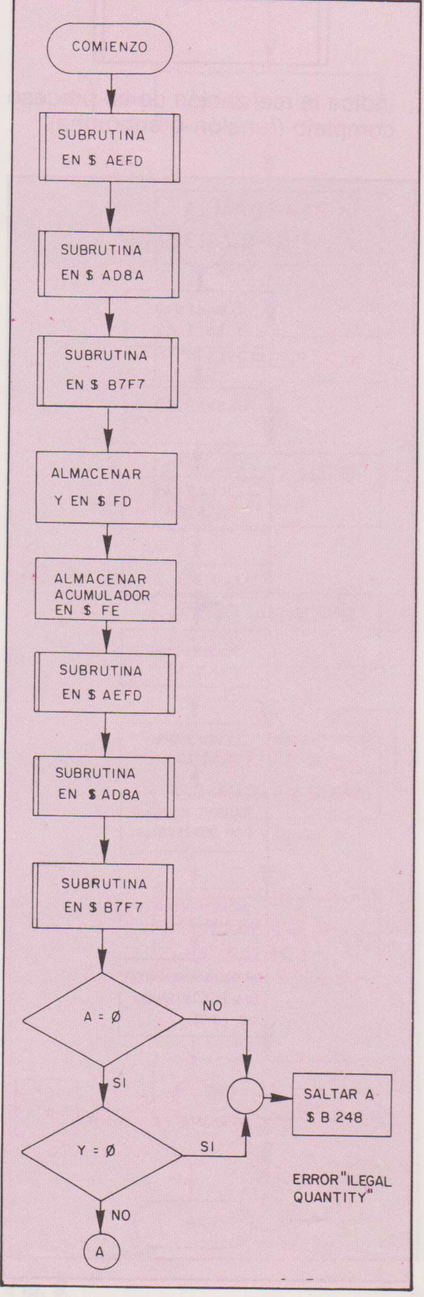
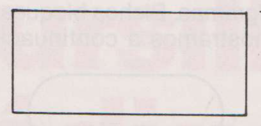
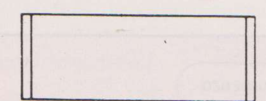


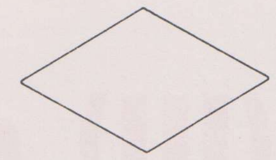
Fig. 3.



: Indica la realización de una instrucción o proceso elemental.

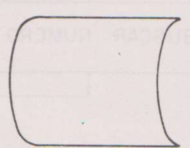


: Indica la realización de un proceso complejo (función o subrutina).



: Indica toma de decisiones.

Además, tenemos los siguientes símbolos auxiliares:



: Indica almacenamiento de datos en memoria secundaria tal como disco, cinta, etc.

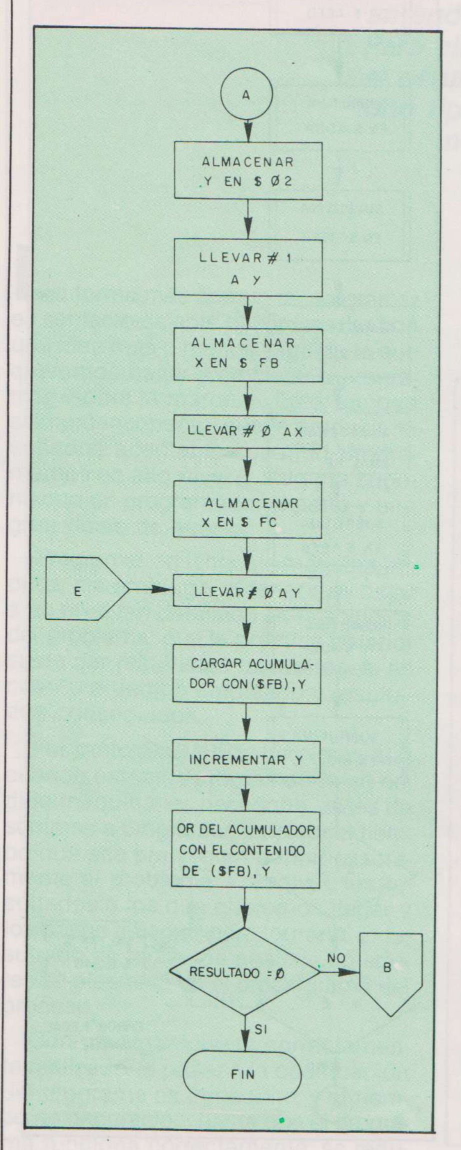


Fig. 4.

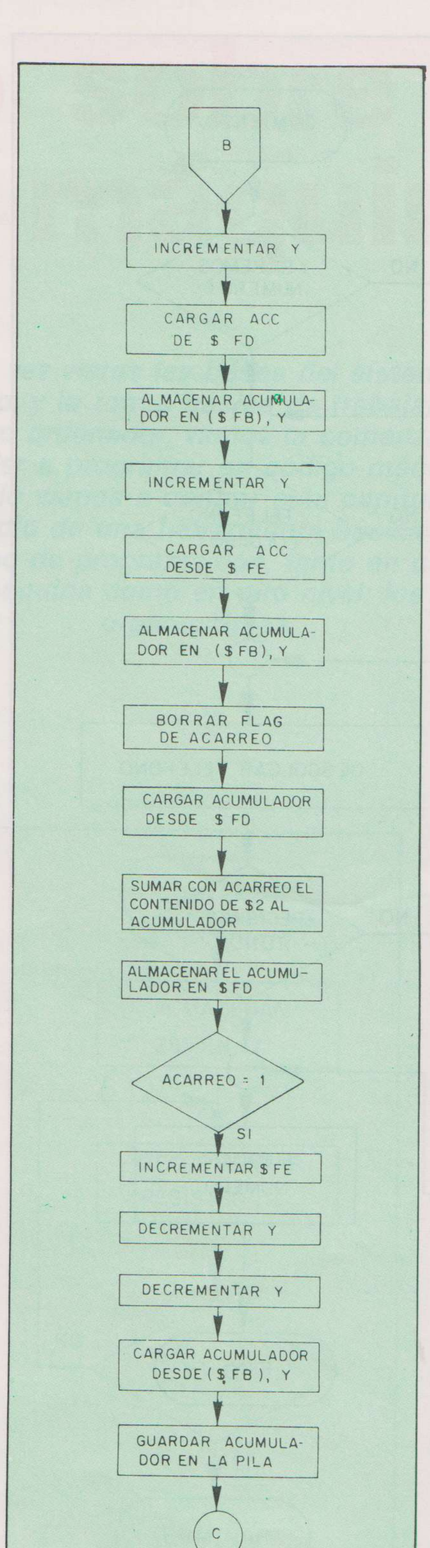


Fig. 5.

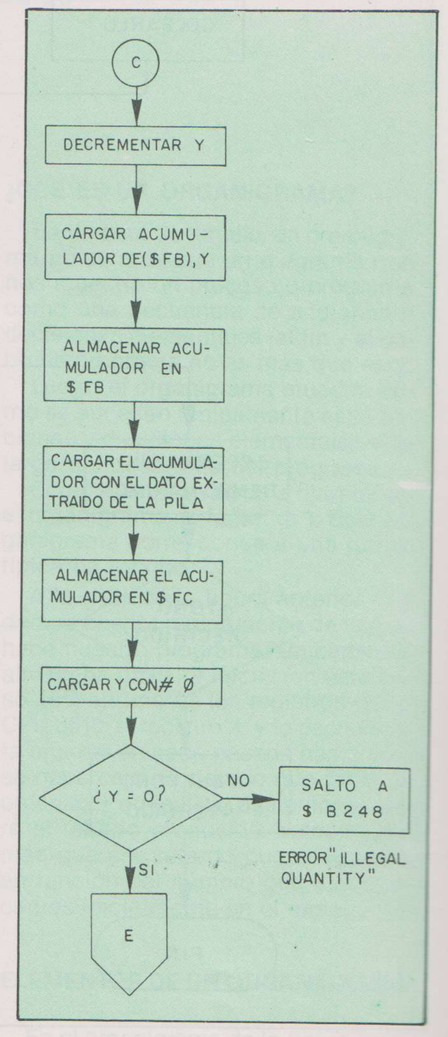
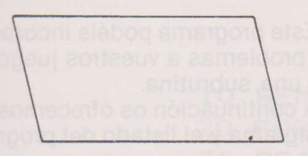
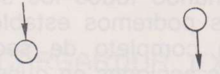


Fig. 6.

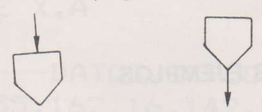


: Operación de entrada/salida en general. Toma de datos, obtención de resultados, etc.

Por último, puede darse el caso de que tengamos que hacer conexiones entre lugares muy distantes del mismo organigrama, bien sea en la misma o en diferente página. Disponemos para ello de los siguientes conectores:



: Conector en la misma página.



: Conector en páginas distintas.

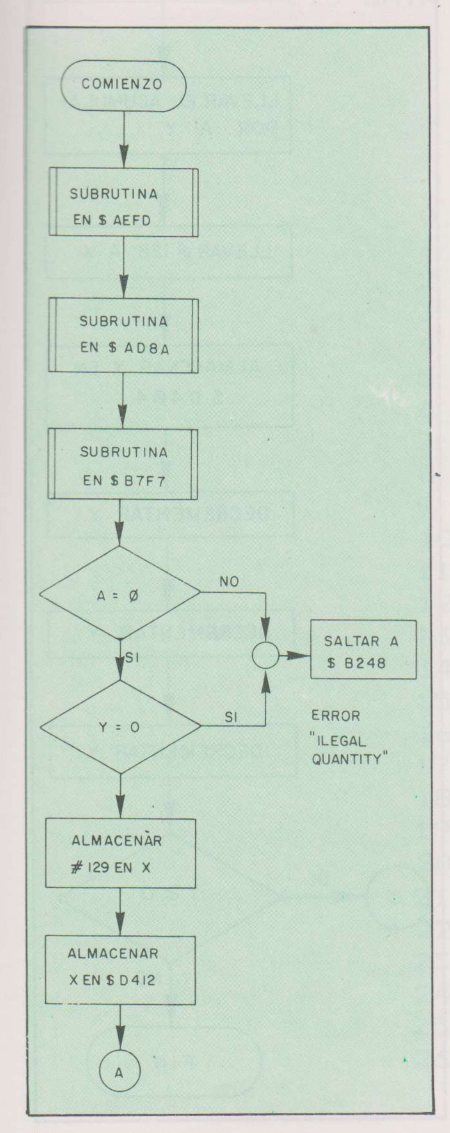


Fig. 7.

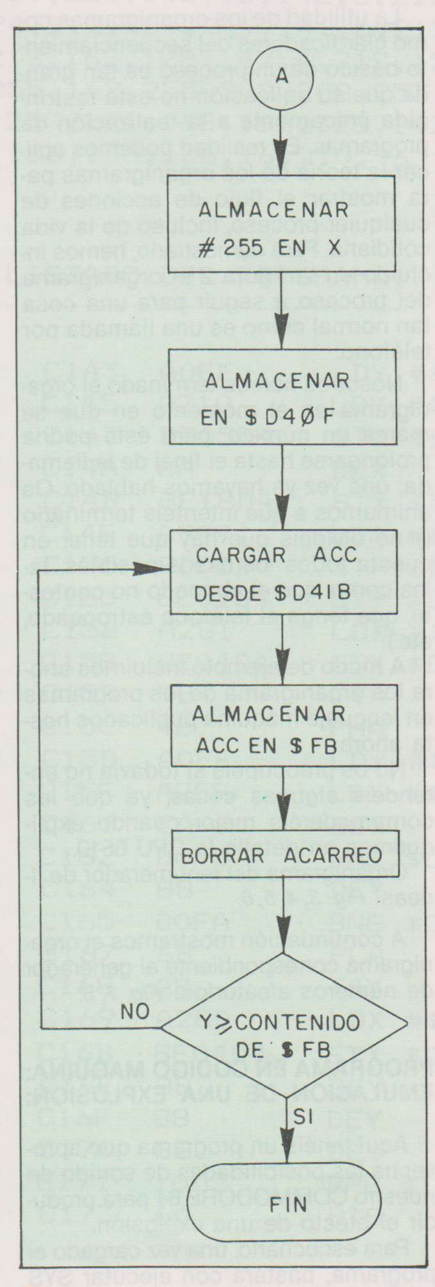


Fig. 8.

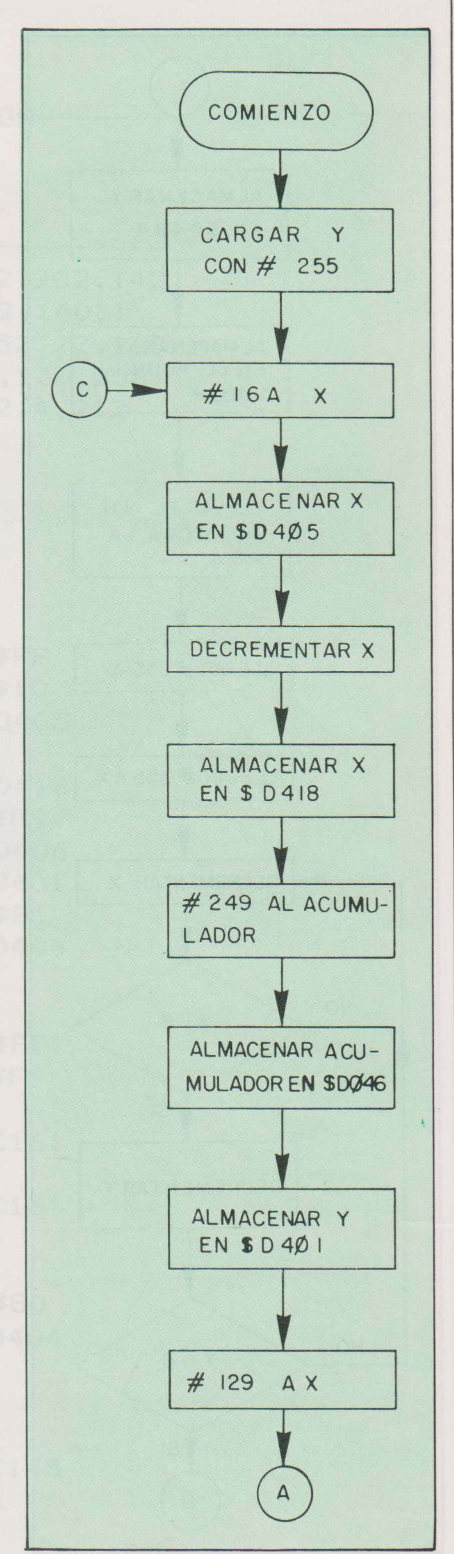


Fig. 9.

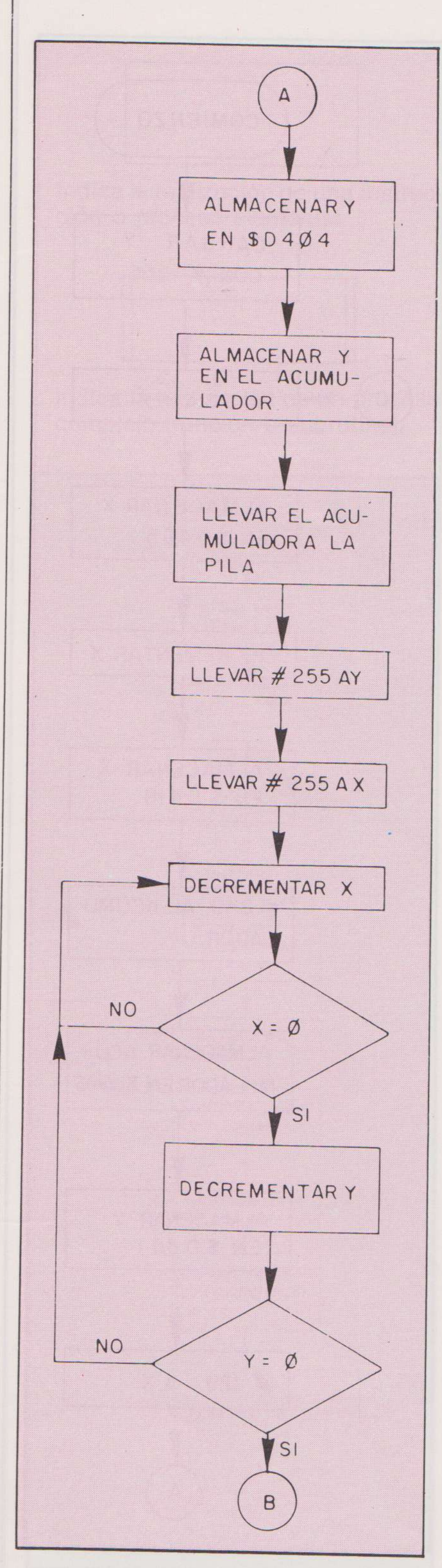


Fig. 10.

Combinando todos los símbolos anteriores podremos establecer un diagrama completo de secuenciamiento de acciones en nuestro programa, tal como hemos visto en la figura 1.

OTROS EJEMPLOS

La utilidad de los organigramas como clarificadores del secuenciamiento básico de un proceso es tan grande que su aplicación no está restringida únicamente a la realización de programas. En realidad podemos aplicar la teoría de los organigramas para mostrar el flujo de acciones de cualquier proceso, incluso de la vida cotidiana. Para demostrarlo, hemos incluido en la figura 2 el organigrama del proceso a seguir para una cosa tan normal como es una llamada por teléfono.

Nosotros hemos terminado el organigrama en el momento en que se marca un número, pero éste podría prolongarse hasta el final de la llamada, una vez ya hayamos hablado. Os animamos a que intentéis terminarlo (y no olvidéis que hay que tener en cuenta todos los casos posibles, tales como, que el abonado no conteste, que tenga el teléfono estropeado, etc.).

A modo de ejemplo incluimos ahora los organigrama de los programas en lenguaje máquina publicados hasta ahora.

No os preocupéis si todavía no entendéis algunas cosas, ya que las comprenderéis mejor cuando expliquemos en detalle la CPU 6510.

Organigrama del renumerador de líneas. *Fig. 3, 4, 5, 6.*

A continuación mostramos el organigrama correspondiente al generador de números aleatorios. *Fig. 7, 8.*

PROGRAMA EN CODIGO MAQUINA: EMULACION DE UNA EXPLOSION:

Aquí tenéis un programa que aprovecha las posibilidades de sonido de nuestro COMMODORE 64 para producir el efecto de una explosión.

Para escucharlo, una vez cargado el programa, bastará con ejecutar SYS 49475.

Este programa podéis incorporarlo sin problemas a vuestros juegos, como una subrutina.

A continuación os ofrecemos el organigrama y el listado del programa: *Fig. 9, 10, 11.*

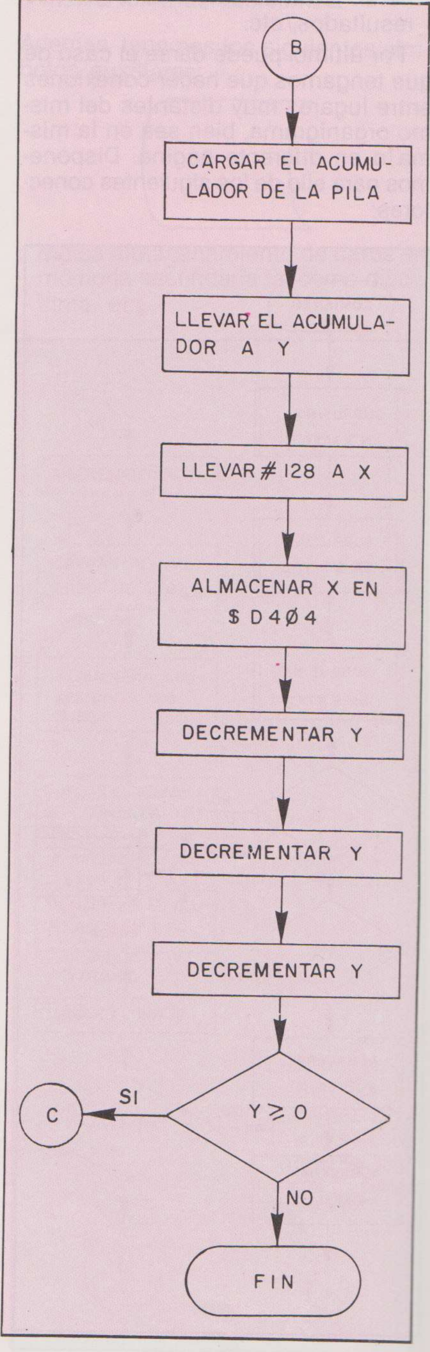


Fig. 11.

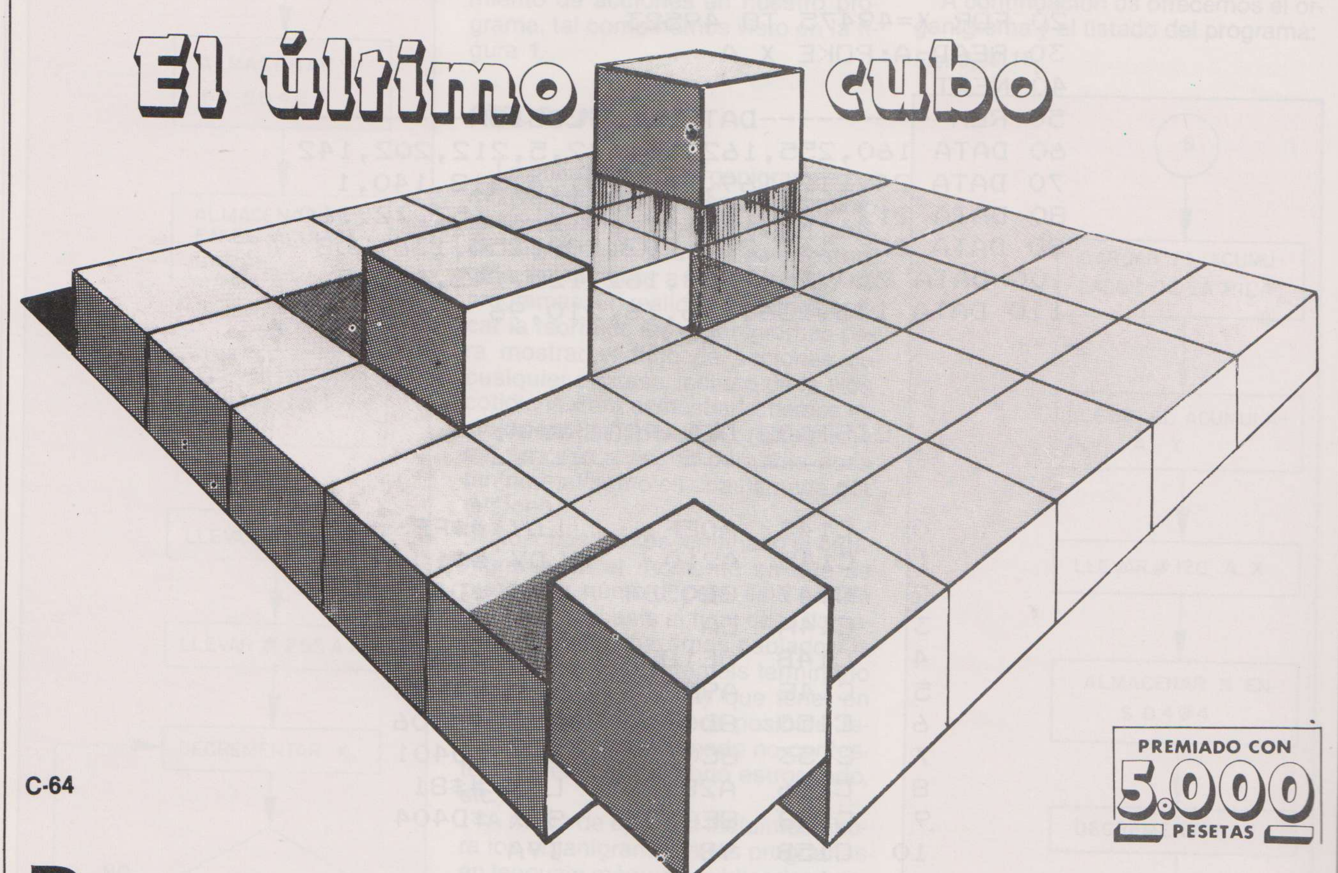
```
10 REM --------CARGARDOR EXPLOSION--------
20 FOR X=49475 TO 49523
30 READ A:POKE X,A
40 NEXT
50 REM ----------DATOS EXPLOSION----------
60 DATA 160,255,162,16,142,5,212,202,142
70 DATA 24,212,169,249,141,6,212,140,1
80 DATA 212,162,129,142,4,212,152,72,160
90 DATA 255,162,255,202,208,253,136,208
100 DATA 250,104,168,162,128,142,4,212
110 DATA 136,136,136,16,210,96
```

LISTADO DEL PROGRAMA:

```
C143  A0FF     LDY ##$FF
C145  A210     LDY ##$10
C147  8E05D4   STX $D405
C14A  CA       DEX
C14B  8E18D4   STX $D418
C14E  A9F9     LDA ##$F9
C150  8D06D4   STA $D406
C153  8C01D4   STY $D401
C156  A281     LDX ##$81
C158  8E04D4   STX $D404
C15B  98       TYA
C15C  48       PHA
C15D  A0FF     LDY ##$FF
C15F  A2FF     LDX ##$FF
C161  CA       DEX
C162  D0FD     BNE $C161
C164  88       DEY
C165  D0FA     BNE $C161
C167  68       PLA
C168  A8       TAY
C169  A280     LDX ##$80
C16B  8E04D4   STX $D404
C16E  88       DEY
C16F  88       DEY
C170  88       DEY
C171  10D2     BPL $C145
C173  60       RTS
```

Concurso

El último cubo



C-64

PREMIADO CON 5.000 PESETAS

Desde Alicante, Fernando Cruz ha remitido a nuestra redacción un programa titulado «El último cubo», un juego muy sencillo que pondrá a prueba la inteligencia de los lectores de Commodore Magazine.

El juego, para el C-64, se puede realizar entre dos personas, o bien, una sola contra el ordenador, pudiéndose escoger en este caso entre dos niveles de dificultad.

También antes de iniciarse el juego, el programa pregunta quién es el jugador que comienza los movimientos (detalle muy importante en este juego, pues el que lo hace goza de una gran ventaja inicial).

Tras escoger las opciones anteriores, aparecen en la pantalla de 15 a 20 cubos, que constituyen el objeto del juego. El juego consiste en que ambos jugadores, en turnos sucesivos y alternativos, retiren de 1 a 3 cubos cada vez. El que retira el último cubo pierde. Si se desea jugar la revancha se puede iniciar una nueva partida.

«El último cubo» resulta un juego interesante, a pesar de ser muy simple, y permite desarrollar y comprobar diferentes estrategias, con lo que ello supone de «ejercicio mental». ¡Y todo en menos de 70 líneas de programa!

```
1 REM * FERNANDO CRUZ - ALICANTE *
5 POKE53280,4:POKE53281,9
8 K$="                                "
9 PRINT"EL ULTIMO CUBO"
10 PRINT"EL JUEGO CONSISTE EN RETIRAR DE 1 A 3"
11 A=INT(RND(0)*(21-15))+15:S=1:X=0::P=0:C$=""
12 PRINT"CUBOS Y QUIEN RETIRE EL ULTIMO PIERDE"
13 PRINT"PULSA '1' JUGAR CONTRA EL ORDENADOR":PRINTSPC(8)"'2' JUGAR ENTRE
DOS"
14 GETP$:IFP$="2"THENP=2:GOTO30
```

```
15 IFP$<>"1"THEN14
16 PRINT"PULSA '1' FACIL        '2' DIFICIL"
17 GETD$:IFD$="1"ORD$="2"THEN19
18 GOTO17
19 D=VAL(D$)
20 PRINT"QUIERES SALIR TU ? (S/N)"
22 GETQ$:IFQ$="S"THEN30
24 IFQ$<>"N"THEN22
28 S=2
30 L=1024:G=55296:PRINT"":FORC=GTOG+999:POKEC,10:NEXT
31 FORC=LTOL+39:POKEC,127:POKEC+800,127:POKEC+960,127:NEXT
32 FORC=LTOL+960STEP40:POKEC,127:POKEC-1,127:NEXT
40 FORV=54272TO54295:POKEV,0:NEXT:PRINT""
42 FORI=1TOA:PRINT"  ___ /  / ‾‾‾‾‾ |   | |  | L_/ TTTT";
44 IFI=7ORI=14THENPRINT""
51 NEXT:IFP=2THEN700
52 IFS=2THEN82
55 IFA<=1THENGOSUB650:PRINTC$"LO SIENTO, HAS PERDIDO":GOSUB600:GOTO200
60 GOSUB650:PRINT  C$TAB(12)"TU RETIRAS ?";:GOSUB800:PRINTX
70 GOSUB300:S=2:GOTO52
82 IFA<=1THENGOSUB650:PRINTC$"BRAVO, HAS GANADO":GOSUB500:GOTO200
83 IFA<=6THEN85
84 IFD=1THENX=INT(RND(0)*3)+1:GOTO95
85 X=1
90 IFA=12ORA=8ORA=4ORA=16THENX=3
92 IFA=11ORA=7ORA=3ORA=15THENX=2
95 GOSUB650
100 PRINTC$TAB(12)"YO RETIRO"X
120 GOSUB300:S=1:GOTO52
200 PRINTC$"PULSA 'S' PARA OTRA PARTIDA   "
210 GETA$:IFA$<>"S"THEN210
220 RUN
300 W=54272:FORI=1TOX:POKEW+24,15
310 IFS=2THENPOKEW,52:POKEW+1,43:GOTO330
320 POKEW,0:POKEW+1,80
330 POKEW+5,9:POKEW+4,0:POKEW+4,33
331 IFA<8THENPRINT""TAB(A*5-3)K$:GOTO336
332 IFA<15THENPRINT""TAB((A-7)*5-3)K$:GOTO336
334 PRINT""TAB((A-14)*5-3)K$
336 A=A-1
340 FORM=1TO800:NEXT:IFA=1THENRETURN
350 NEXT:RETURN
500 FORV=54272TO54295:POKEV,0:NEXT
505 POKEW+5,3:POKEW+4,33
510 FORD=1TO200:NEXT:READA,B
520 IFB=-1THENPOKEW+1,0:POKEW,0:RESTORE:RETURN
530 POKEW+1,A:POKEW,B:POKEW+4,0:GOTO505
540 DATA17,37,19,63,21,154,22,227,17,37,63,154,22,227,52,177,28,214,32,94,34,175
550 DATA-1,-1
600 FORV=54272TO54295:POKEV,0:NEXT
610 POKEW+6,240:POKEW+1,4:POKEW+5,0:POKEW+4,33:FORH=1000TO300STEP-50
620 POKEW+1,H/256:POKEW,HAND255:FORR=1TO200:NEXTR,H:POKEW+4,0:RETURN
650 PRINTC$"                              ":RETURN
700 IFA<=1THENGOSUB650:PRINTC$"HA GANADO EL SEGUNDO JUGADOR":GOSUB500:GOTO2
00
710 GOSUB650:PRINTC$"  EL PRIMER JUGADOR RETIRA ?";:GOSUB800:PRINTX
730 S=2:GOSUB300
750 IFA<=1THENGOSUB650:PRINTC$"HA GANADO EL PRIMER JUGADOR":GOSUB500:GOTO20
0
760 GOSUB650:PRINTC$"  EL SEGUNDO JUGADOR RETIRA ?";:GOSUB800:PRINTX
780 S=1:GOSUB300:GOTO700
800 GETX$:X=VAL(X$):IFX<1ORX>3THEN800
810 RETURN
```

PROGRAMA commodore Magazine COMPROBADO

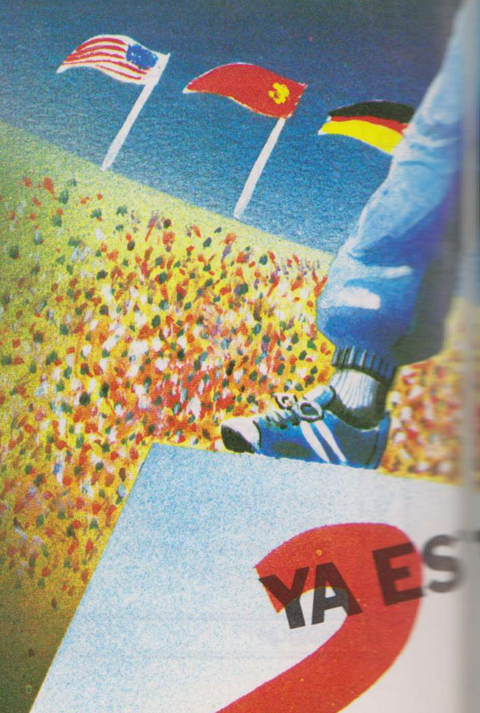
SUMMER

Para COMM

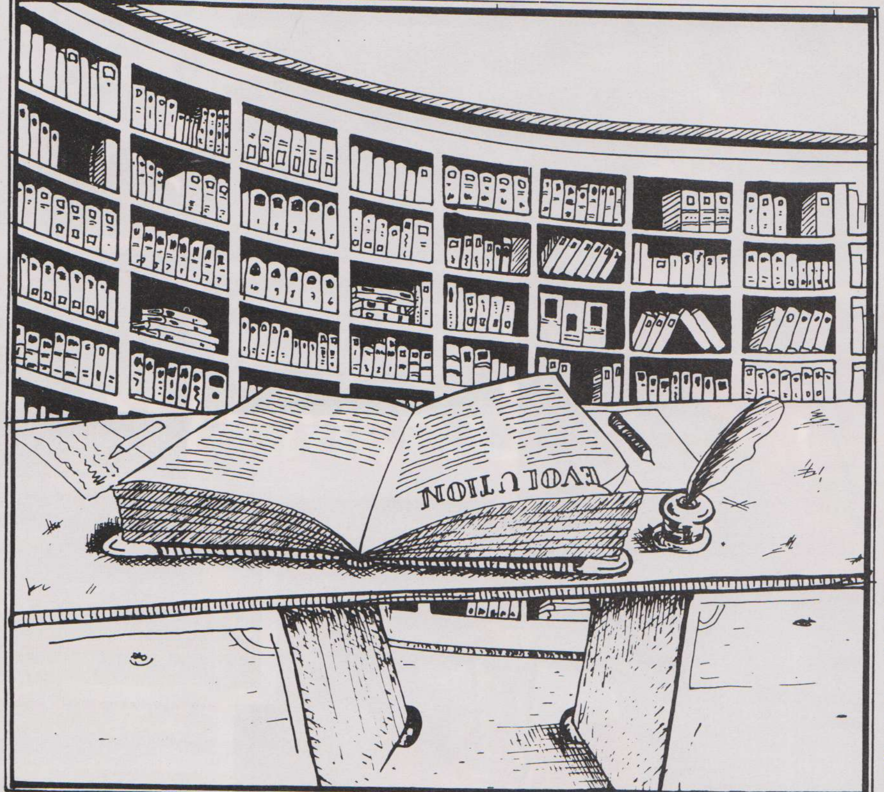


- Ocho nuevos eventos, in
  - Ciclismo.
  - Esgrima.
  - Kaiak.
  - Triple salto.
  - Remo.
  - Salto de altura.
  - Jabalina.
  - Equitación.
- Ceremonia de apertura, cie
  con himnos nacionales.
- Compita contra el ordena-
  dor o contra sus amigos
  o familiares.
- Competición indivi-
  dual o codo a codo.
- De uno a ocho
  jugadores.
- Control joystick.





Concurso



CBM 64

Biblioteca 64

A todos aquellos de nuestros lectores que posean una pequeña biblioteca casera les será de gran utilidad el programa BIBLIOTECA-64, remitido a nuestro concurso por Eduardo Casino, desde Santander, que constituye un excelente medio para llevar el control de todos los libros.

El programa sirve para guardar ordenadamente las fichas correspondientes a los libros que forman la biblioteca. Cada ficha tiene los siguientes campos de información:

— Autor: Apellidos
Nombre
Nacionalidad

— Libro: Título
Tema
Editorial
Fecha

Luego el programa añade un número de ficha distinto.

Al inciar el programa aparece un pre-menú con dos opciones:

(1) Abrir un nuevo fichero: comienza un fichero vacío.

(2) Trabajar sobre el anterior: permite confeccionar, corregir o anular fichas de un fichero ya creado.

Si se escoge la opción (1) se pasa directamente a la confección de las nuevas fichas. Si se sigue la opción (2) hay que cargar previamente el fichero de que se trate desde el cassette, y a continuación se pasa al menú principal con las siguientes opciones:

(1) Confeccionar fichas.
(2) Lectura (búsqueda) de fichas.
(3) Corrección de fichas.
(4) Anulación de fichas.
(5) Vuelta al pre-menú.
(6) Ordenación alfabética de las fichas.
(7) Grabación de los datos en cassette.
(8) Memoria disponible.
(9) Finalización.

Todas las opciones son autoexplicativas y su utilización es sencilla. Sólo cabe resaltar la opción (2), que realiza la búsqueda de una ficha a partir de la información de uno de sus campos. Así, por ejemplo, esto puede servir para ver todos los libros que se posee de un autor determinado, o los libros editados en un año cualquiera, o los que tratan de un tema concreto. La opción (6), de ordenación alfabética, permite ordenar las fichas por autores, títulos o temas, lo que resulta suficiente para una biblioteca casera.

Sólo hay que tener en cuenta la limitación en la extensión de las informaciones de cada campo, pues el programa no admite más que un cierto número de caracteres, de todas formas el propio programa avisa cuando se sobrepasa dicho límite.

A continuación se expone la estructura general del programa:

1-10	Presentación.
11-25	Dimensionamiento e inicialización de variables.
26-34	Pre-menú.
35-50	Carga de datos del fichero anterior.
51-66	Menú principal.
67-90	Confección de fichas.
91-199	Lectura de fichas.
200-256	Corrección de fichas.
257-274	Anulado de fichas.
275-293	Grabación de datos.
294-319	Ordenación alfabética de fichas.
321-324	Cálculo de la memoria disponible.
325-327	Aviso de número excesivo de caracteres.

```
1 REM ***********************************
2 REM **                              **
3 REM **        BIBLIOTECA-64         **
4 REM **                              **
5 REM ***********************************
6 REM **                              **
7 REM **     POR EDUARDO CASINO       **
8 REM **                              **
9 REM ***********************************
10 :
11 REM ** DIMENSIONADO DE VARIABLES **
12 :
13 DIMT$(100),TE$(100),NA$(100),AA$(100),NC$(100),E$(100),F$(100),G$(100)
14 DIMF(100),C(100),S$(100)
15 :
16 REM ** INICIALIZACION VARIABLES **
17 :
18 A$="                                              BIBLIOTECA-64"
19 B$="                                              "
20 A$=A$+B$
21 XX$="                                        "
22 W$="                                        "
23 :
24 POKE53281,0:POKE53280,0
25 :
26 REM **          PRE-MENU          **
27 :
28 PRINTA$:PRINT"                 PRE-MENU":PRINT"                    "
29 PRINT"   * ABRIR UN NUEVO FICHERO........(1)
30 PRINT"   * TRABAJAR SOBRE EL ANTERIOR....(2)"
31 GET B$:IF B$=""THEN31
32 IFB$="1"THEN69
33 IFB$<>"2"THEN31
34 :
35 REM ** CARGA DATOS DEL FICHERO **
36 :
37 PRINTA$
38 PRINT".-REBOBINA LA CINTA HASTA EL INICIO DEL  FICHERO"
39 PRINT"  DESPUES PULSA 'P' Y SIGUE         INSTRUCCIONES.
40 PRINT"  'V' VUELVE AL PRE-MENU."
41 GETB$:IFB$=""THEN41
42 IFB$="V"THEN28
43 IFB$<>"P"THEN41
44 OPEN1,1,0,"BIBLIOFICH"
45 INPUT#1,ND
46 FORA=1TOND
47 INPUT#1,T$(A):INPUT#1,TE$(A):INPUT#1,NA$(A):INPUT#1,AA$(A)
48 INPUT#1,NC$(A):INPUT#1,E$(A):INPUT#1,F$(A)
49 NEXTA:CLOSE1
50 :
51 REM **         MENU         **
52 :
53 PRINTA$:PRINT"                 MENU":PRINT"                    "
54 PRINT"   * CONFECCIONAR FICHAS..............(1)"
55 PRINT"   * LECTURA DE FICHAS................(2)"
56 PRINT"   * CORRECCION DE FICHAS.............(3)"
57 PRINT"   * ANULADO DE DATOS.................(4)"
58 PRINT"   * VUELTA AL PRE-MENU...............(5)"
59 PRINT"   * ORDEN ALF. FICHAS................(6)"
60 PRINT"   * GRABACION DE DATOS...............(7)"
61 PRINT"   * MEMORIA DISPONIBLE...............(8)"
62 PRINT"   * FINALIZAR........................(9)"
```

```
63 GETB$:IF B$="" ORVAL(B$)<1ORVAL(B$)>9THEN63
64 IFB$="9"THENPRINTA$:END
65 ONVAL(B$)GOTO69,93,202,259,28,293,277,321
66 :
67 REM **    CONFECCION DE FICHAS     **
68 :
69 PRINTA$: PRINT"             ■CONFECCION DE FICHAS"
70 PRINT"          ■──────────────────"
71 PRINT"■          ■(INTRODUCIENDO ■-1■ VUELVE                        AL MENU)
72 ND=ND+1
73 INPUT"■.-TITULO:  ■";T$(ND):C$=T$(ND):GOSUB88
74 IFLEN(T$(ND))>30THENPRINTA$:GOSUB325:GOTO73
75 INPUT"■■.-TEMA:  ■";TE$(ND):C$=TE$(ND):GOSUB88
76 IFLEN(TE$(ND))>32THENPRINTA$:GOSUB325:GOTO75
77 INPUT"■■.-NOMBRE AUT.:  ■";NA$(ND):C$=NA$(ND):GOSUB88
78 IFLEN(NA$(ND))>29THENPRINTA$:GOSUB325:GOTO77
79 INPUT"■■.-APELLIDOS:  ■";AA$(ND):C$=AA$(ND):GOSUB88
80 IFLEN(NA$(ND))+LEN(AA$(ND))>29THENPRINTA$:GOSUB325:GOTO79
81 INPUT"■■.-NACIONALIDAD:  ■";NC$(ND):C$=NC$(ND):GOSUB 88
82 IFLEN(NC$(ND))>24THENPRINTA$:GOSUB325:GOTO81
83 INPUT"■■.-EDITORIAL:  ■";E$(ND):C$=E$(ND):GOSUB88
84 IFLEN(E$(ND))>27THENPRINTA$:GOSUB325:GOTO83
85 INPUT"■■.-FECHA DE ED.:  ■";F$(ND):C$=F$(ND):GOSUB88
86 IFLEN(F$(ND))>27THENPRINTA$:GOSUB325:GOTO85
87 GOTO69
88 IFC$="-1"THENND=ND-1:GOTO53
89 RETURN
90 :
91 REM **    LECTURA DE LAS FICHAS    **
92 :
93 PRINTA$:PRINT"■                 LECTURA DE FICHAS"
94 PRINT"■                 ─────────────────"
95 IFND<>0THEN100
96 PRINT"■■   ■■■ NO HAY NINGUN DATO."
97 PRINT"■■    ■('M' VUELVE AL MENU)"
98 GETB$:IFB$<>"M"THEN98
99 GOTO53
100 PRINT"    ■TABLA DE POSIBILIDADES DE BUSQUEDA■■"
101 PRINT"  ┌─┬──────────┬──────■ ■                 ┐"
102 FORA=1TO4
103 PRINT"  │ │          │      ■ ■                 │":NEXTA
104 PRINT"  │ │          │      ■                   "
105 PRINT"  └─┴──────────┴──────■ ■                 "
106 PRINT"  │ │          │      ■ ■ ┌─────────────┐ │"
107 FORA=1TO6
108 PRINT"  │ │          │      ■ ■               │":NEXTA
109 PRINT"  │ │          │      ■ ■               │"
110 PRINT"  └─┴──────────┴──────■ ■───────────────┘"
111 PRINT"■■■■■■■■■■■■■■■■A■APELLIDOS■■■■■NUM. DE FICHA■■■6
112 PRINT"■■■■■■■■■■■■■■■■■■■■NOMBRE■■■■2■■TODAS■■■■■■■■7"
113 PRINT"■■■■■■■■■■■■■■■■■■■■D.■■NACIONAL.■■■3"
114 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■TITULO■■■■■4"
115 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■TEMA■■■■■■5"
116 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■B■■■EDITORIAL■■6"
117 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■D.■■FECHA ED.■■7";
118 PRINT"■■■■■■■■■■■'ESPACIO' VA■■■■■■■■■■■■AL MENU"
119 IFB$=" "THEN53
120 GETB$:IFB$=""ORB$=" "ORVAL(B$)<1ORVAL(B$)>9THEN117
121 IFB$=" "THEN53
122 B=VAL(B$)
123 PRINTA$:PRINT"               ■LECTURA DE FICHAS"
124 PRINT"               ■─────────────────"
```

PROGRAMA commodore Magazine COMPROBADO

Atrévete con la nueva SG 10 Commodore de STAR

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←
```

Con la misma impresora podrás trabajar directamente con tu Commodore y si algún día te atreves con el PC de Commodore la misma impresora te servirá sólo cambiando un interruptor.
Las impresoras STAR te ofrecen: 120 cps., 100 tipos de letra diferentes, letra de calidad (NLQ), fricción-tracción, cinta de máquina de escribir, si trabajas con el Simon's Basic podrás hacer hard copys directamente de pantalla.
El futuro está en el universo y STAR es tu estrella.

De venta en establecimientos especializados:

IMPORTADO POR:



COMPONENTES ELECTRONICOS, S.A.

08009 BARCELONA. Consejo de Ciento, 409
Tel. (93) 231 59 13

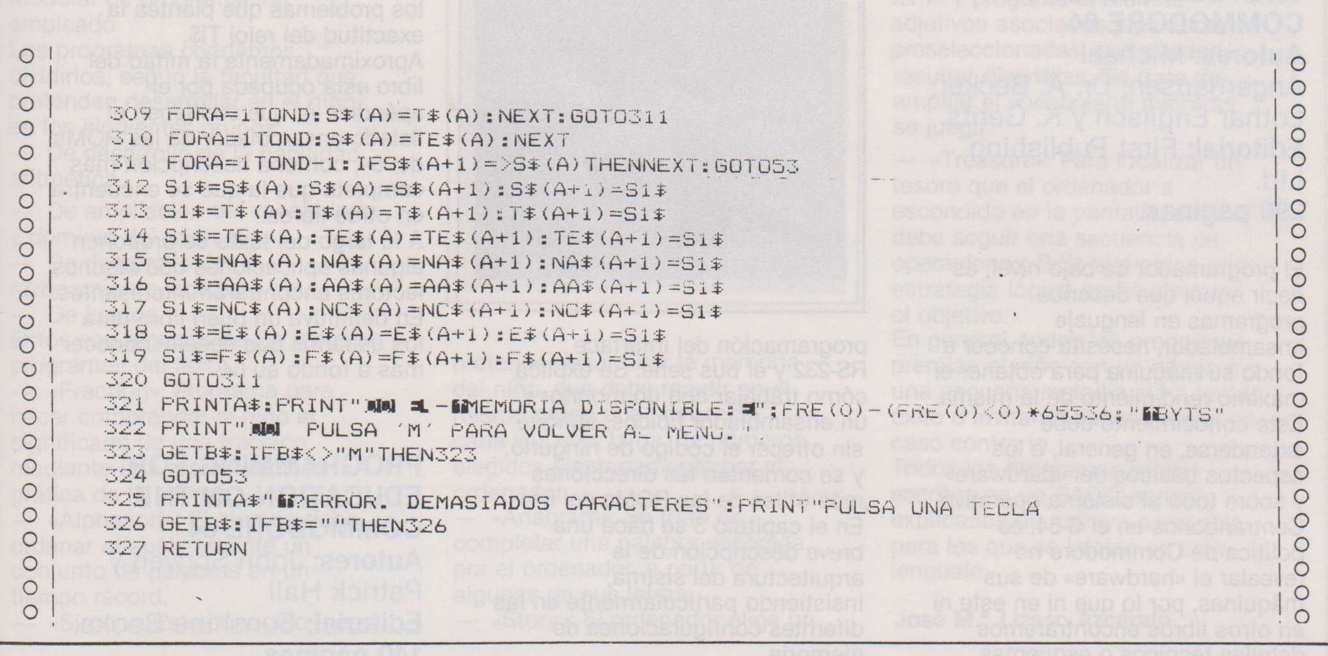
28020 MADRID. Comandante Zorita, 13
Tels. (91) 233 00 94 - 233 09 24

```
125 ONBGOTO126,129,132,135,138,141,144,147,153
126 INPUT"▓▓.-APELLIDOS: ▓";D$
127 FORA=1TOND:G$(A)=AA$(A):NEXTA
128 GOTO154
129 INPUT"▓▓.-NOMBRE: ▓";D$
130 FORA=1TOND:G$(A)=NA$(A):NEXTA
131 GOTO154
132 INPUT"▓▓.-NACIONALIDAD: ▓";D$
133 FORA=1TOND:G$(A)=NC$(A):NEXTA
134 GOTO154
135 INPUT"▓▓.-TITULO: ▓";D$
136 FORA=1TOND:G$(A)=T$(A):NEXT A
137 GOTO154
138 INPUT"▓.-TEMA: ▓";D$
139 FORA=1TOND:G$(A)=TE$(A):NEXTA
140 GOTO154
141 INPUT"▓.-EDITORIAL: ▓";D$
142 FORA=1TOND:G$(A)=E$(A):NEXTA
143 GOTO154
144 INPUT"▓.-FECHA DE EDICION: ▓";D$
145 FORA=1TOND:G$(A)=F$(A):NEXTA
146 GOTO154
147 D=0:PRINT"  ▓▓ INTRODUZCA LOS NUMEROS DE LAS FICHAS    QUE QUIERA VER. "
148 PRINT"   ▓▓ 0▓ FINALIZA▓▓"
149 D=D+1:INPUT"▓▓.-NUMERO:     ▓▓▓";C(D):IFC(D)=0THEND=D-1:GOTO53
150 IFC(D)<0ORC(D)>NDTHEND=D-1:PRINT"▓.-▓NO EXISTE.▓"
151 GOTO149
152 FORS=1TOD:A=C(S):GOTO163
153 FORT=1TOND:A=T:GOTO163
154 E=0:FORA=1TOND
155 IFLEFT$(D$,LEN(D$))=LEFT$(G$(A),LEN(D$))THEN E=E+1:F(E)=A
156 NEXT:IFE=0THEN158
157 GOTO162
158 PRINT"▓▓ ▓▓ NO EXISTE NINGUNA FICHA CON            ESOS DATOS."
159 PRINT"     ('M' VUELVE AL MENU.)"
160 GET B$:IFB$<>"M"THEN160
161 GOTO53
162 FORR=1TOE:A=F(R)
163 PRINTA$:PRINT"            ▓LECTURA DE FICHAS"
164 PRINT"            ▓              "
165 PRINT"┌──────────────────────────────────────┐";
166 PRINT"│ ▓NUM. ▓"STR$(A);SPC(40-(LEN(STR$(A))+6))" │";
167 PRINT"│ "SPC(38)" │";
168 PRINT"│ ▓TITULO: ▓"T$(A);SPC(40-(LEN(T$(A))+10))" │";
169 PRINT"│ "SPC(38)" │";
170 PRINT"│ ▓TEMA: ▓"TE$(A);SPC(40-(LEN(TE$(A))+8))" │";
171 PRINT"│ "SPC(38)" │";
172 PRINT"│ ▓AUTOR: ▓"AA$(A)", "NA$(A);SPC(40-(LEN(AA$(A)+NA$(A))+11))" │";
173 PRINT"│ "SPC(38)" │";
174 PRINT"│ ▓NACIONALIDAD: ▓"NC$(A);SPC(40-(LEN(NC$(A))+16))" │";
175 PRINT"│ "SPC(38)" │";
176 PRINT"│ ▓EDITORIAL: ▓"E$(A);SPC(40-(LEN(E$(A))+13))" │";
177 PRINT"│ "SPC(38)" │";
178 PRINT"│ ▓FECHA ED.: ▓"F$(A);SPC(40-(LEN(F$(A))+13))" │";
179 PRINT"│ "SPC(38)" │";
180 PRINT"└──────────────────────────────────────┘"
181 PRINT"'M' VUELVE AL MENU. 'L' VUELVE A LECTURA.";
182 IFB<>8THEN185
183 IFS=DTHEN196
184 GOTO189
185 IFB<>9THEN188
186 IFT=NDTHEN196
```

```
187 GOTO189
188 IFR=ETHEN196
189 PRINT"'C' CONTINUA.";
190 GETB$:IFB$="M"THEN53
191 IFB$="L"THEN196
192 IFB$<>"C"THEN190
193 IFB=8THENNEXTS
194 IFB=9THENNEXTT
195 NEXTR
196 GETB$:IFB$="M"THEN53
197 IFB$<>"L"THEN196
198 GOTO92
199 :
200 REM **   CORRECCION DE FICHAS   **
201 :
202 H$=A$+"                                        "
203 H$=H$+"         CORRECCION DE FICHAS"
204 H$=H$+"  ________________________________":PRINTH$
205 IFND=0THENPRINT"  NO HAY NINGUN DATO. 'M' VUELVE        AL MENU":GOTO207
206 GOTO209
207 GETB$:IFB$<>"M"THEN207
208 GOTO53
209 PRINT"  0 VUELVE AL MENU.":INPUT"  NUMERO DE FICHAS A CORREGIR";NF
210 IFNF>NDTHENNF=ND:GOTO213
211 IFNF=0THEN53
212 IFNF<0THEN202
213 FORA=1TONF:PRINTH$
214 PRINT"     -1 VA A LECTURA DE DATOS.
215 PRINT"      0 VUELVE AL MENU.
216 INPUT"  NUMERO DE LA FICHA";FF:IFFF>NDTHENPRINT"NO EXISTE.":GOTO216
217 IFFF=0THEN53
218 IFFF=-1THEN93
219 IFFF<-1THEN213
220 PRINTH$
221 PRINT"  SI EL DATO ES CORRECTO, PULSA 'RETURN'   SI NO LO ES, CORRIGELO.
222 PRINT"  TITULO:  ";T$(FF)
223 PRINT"  TEMA:  ";TE$(FF)
224 PRINT"  NOMBRE AUT.:  ";NA$(FF)
225 PRINT"  APELLIDOS:  ";AA$(FF)
226 PRINT"  NACIONALIDAD:  ";NC$(FF)
227 PRINT"  EDITORIAL:  ";E$(FF)
228 PRINT"  FECHA ED.:  ";F$(FF)
229 X$="":INPUT"  TIT.  ";X$:IFX$=""THEN233
230 IFLEN(X$)>30THENGOSUB325:GOTO220
231 PRINT"                    "X$;LEFT$(W$,40-(LEN(X$)+9));"              ";
232 T$(FF)=X$:PRINTXX$
233 X$="":INPUT"  TEM.  ";X$:IFX$=""THEN236
234 PRINT"                    "X$;LEFT$(W$,40-(LEN(X$)+7));"            ";
235 TE$(FF)=X$:PRINTXX$
236 X$="":INPUT"  NOM.  ";X$:IFX$=""THEN239
237 PRINT"                            "X$;LEFT$(W$,40-(LEN(X$)+14));"        ";
238 NA$(FF)=X$:PRINTXX$
239 X$="":INPUT"  APE.  ";X$:IFX$=""THEN242
240 PRINT"                            "X$;LEFT$(W$,40-(LEN(X$)+12));"        ";
241 AA$(FF)=X$:PRINTXX$
242 X$="":INPUT"  NAC.  ";X$:IFX$=""THEN245
243 PRINT"                                    "X$;LEFT$(W$,40-(LEN(X$)+15));"    ";
244 NC$(FF)=X$:PRINTXX$
245 X$="":INPUT"  EDI.  ";X$:IFX$=""THEN248
246 PRINT"                                    "X$;LEFT$(W$,40-(LEN(X$)+12));"   ";
247 E$(FF)=X$:PRINTXX$
248 X$="":INPUT"  FE .  ";X$:IFX$=""THEN251
249 PRINT"                                    "X$;LEFT$(W$,40-(LEN(X$)+8));"  ";
250 F$(FF)=X$:PRINTXX$
```

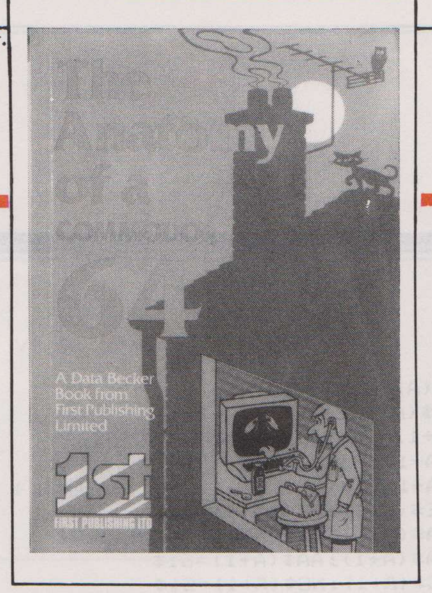

PRIMERA REVISTA EN CASSETTE
Nº5
64
commodore
TAPE
COMPUTING
Con VOZ digitalizada
REVISTA INDEPENDIENTE EN CASSETTE
PARA USUARIOS DEL COMMODORE 64
APLICACIONES
SUPER BASE DE DATOS
Genere sus propias aplicaciones
EDUCATIVOS
ARITMETICA
Escuche las protestas del profesor
MULTICOLOR
Dibuje en alta resolución multicolor
UTILIDADES
SUPER BASIC
20 Nuevos comandos para programar más fácilmente
JUEGOS
GOLDRUSH
GRID BASH
LASER CYCLES
Y MUCHO MAS
BIENVENIDO A
64 commodore TAPE COMPUTING
PRESENTA
VIDEO DIGITALIZADOR
FICHERO AYUDA
ARITMETICA
SCROLLING
GOLD RUSA
JOYSTICK AID
MULTICOLOR
GRID BASH
SUPER BASIC
LASER CYCLES
SLIPPERY SID
SUPER BASE DE DATOS
commodore
¡YA ESTA A LA VENTA EN SU KIOSKO!
875 Ptas
Solicítela a: INFODIS C/ Bravo Murillo, 377 - 5.º A - 28020 MADRID
CUPON DE PEDIDO
Sí, envíenme al precio de 875 ptas.... ejs. de 64 TAPE COMPUTING Nº5
El importe lo abonaré: Contra reembolso ☐ Con mi tarjeta de crédito ☐
Adjunto cheque ☐ American Express ☐ Visa ☐ Interbank ☐
Número de mi tarjeta
Fecha de caducidad
NOMBRE
DIRECCION
CIUDAD
PROVINCIA
Sin gastos de envío

```
251 PRINT"]'S' SIGUE. 'M' VA AL MENU."
252 GETB$:IFA=NFTHEN254
253 IFB$="S"THENNEXT
254 IFB$<>"M"THEN252
255 GOTO53
256 :
257 REM **      ANULADO DE FICHAS     **
258 :
259 PRINTA$:PRINT"                  ANULADO DE DATOS"
260 PRINT"            ─────────────────────"
261 IFND<>0THEN265
262 PRINT" -NO HAY NINGUN DATO. 'M' VELVE            AL MENU."
263 GETB$:IFB$<>"M"THEN263
264 GOTO53
265 PRINT" -INTRODUZCA LOS NUMEROS DE LAS FICHAS     QUE DESEE ANULAR."
266 PRINT" -0 VUELVE AL MENU. -1 VA A LECTURA        DE DATOS."
267 PRINT"** LOS NUMEROS DEBEN SER INTRODUCIDOS       DE MAYOR A MENOR!!!!!!"
268 INPUT"] NUM. DE FICHA :";NF
269 IFNF=0THEN53
270 IFNF=-1THEN93
271 IFNF>NDORNF<-1THENPRINT"NO EXISTE.":GOTO268
272 FORA=NFTOND:T$(A)=T$(A+1):TE$(A)=TE$(A+1):NA$(A)=NA$(A+1):AA$(A)=AA$(A+1)
273 NC$(A)=NC$(A+1):E$(A)=E$(A+1):F$(A)=F$(A+1):NEXTA:ND=ND-1:GOTO268
274 :.
275 REM **    GRABACION DE DATOS     **
276 :
277 PRINTA$:PRINT"              GRABACION DE DATOS'
278 PRINT"           ─────────────────────
279 IFND<>0THEN283
280 PRINT".-NO HAY NINGUN DATO. 'M' VELVE AL MENU."
281 GETB$:IFB$<>"M"THEN281
282 GOTO53
283 PRINT"** REBOBINA LA CINTA HASTA ''EL INICIO      DEL FICHERO'':!!"
284 PRINT"** DESPUES PULSA 'G' PARA COMENZAR           LA GRABACION. 'M' VA MEN
U "
285 GETB$:IFB$="M"THEN53
286 IFB$<>"G"THEN285
287 OPEN1,1,1,"BIBLIOFICH"
288 PRINT#1,ND
289 FORA=1TOND
290 PRINT#1,T$(A):PRINT#1,TE$(A):PRINT#1,NA$(A):PRINT#1,AA$(A):PRINT#1,NC$(A)
291 PRINT#1,E$(A):PRINT#1,F$(A)
292 NEXT:CLOSE1:GOTO53
293 :
294 REM **   ORDEN ALF. DE FICHAS     **
295 :
296 PRINTA$:PRINT"                  ORDEN ALF. FICHAS":PRINT"              ─────────────
────"
297 IFND>0THEN301
298 PRINT" -NO HAY NINGUN DATO. 'M' VA AL MENU."
299 GETB$:IFB$<>"M"THEN299
300 GOTO53
301 PRINT" -ORDENO POR AUTORES, POR OBRAS O POR      TEMAS (A/O/T)?"
302 PRINT"  'M' VA AL MENU."
303 GETB$:IFB$="A"THEN308
304 IFB$="O"THEN309
305 IFB$="T"THEN310
306 IFB$<>"M"THEN303
307 GOTO53
308 FORA=1TOND:S$(A)=AA$(A)+NA$(A):NEXT:GOTO311
```



```
309 FORA=1TOND:S$(A)=T$(A):NEXT:GOTO311
310 FORA=1TOND:S$(A)=TE$(A):NEXT
311 FORA=1TOND-1:IFS$(A+1)=>S$(A)THENNEXT:GOTO53
312 S1$=S$(A):S$(A)=S$(A+1):S$(A+1)=S1$
313 S1$=T$(A):T$(A)=T$(A+1):T$(A+1)=S1$
314 S1$=TE$(A):TE$(A)=TE$(A+1):TE$(A+1)=S1$
315 S1$=NA$(A):NA$(A)=NA$(A+1):NA$(A+1)=S1$
316 S1$=AA$(A):AA$(A)=AA$(A+1):AA$(A+1)=S1$
317 S1$=NC$(A):NC$(A)=NC$(A+1):NC$(A+1)=S1$
318 S1$=E$(A):E$(A)=E$(A+1):E$(A+1)=S1$
319 S1$=F$(A):F$(A)=F$(A+1):F$(A+1)=S1$
320 GOTO311
321 PRINTA$:PRINT"██ ▀▄-▜MEMORIA DISPONIBLE:▀";FRE(0)-(FRE(0)<0)*65536;"▜BYTS"
322 PRINT"██  PULSA 'M' PARA VOLVER AL MENU."
323 GETB$:IFB$<>"M"THEN323
324 GOTO53
325 PRINTA$"▜?ERROR. DEMASIADOS CARACTERES":PRINT"PULSA UNA TECLA
326 GETB$:IFB$=""THEN326
327 RETURN
```

Libros



THE ANATOMY OF A COMMODORE 64

Autores: Michael Angerhausen, Dr. A. Becker, Lothar Englisch y K. Gerits.
Editorial: First Publishing Ltd.
290 páginas.

El programador de bajo nivel, es decir aquél que describe programas en lenguaje ensamblador, necesita concoer a fondo su máquina para obtener el máximo rendimiento de la misma. Este conocimiento debe extenderse, en general, a los aspectos básicos del «hardware» y sobre todo al sistema operativo.
Centrándonos en el C-64, es política de Commodore no revealar el «hardware» de sus máquinas, pcr lo que ni en este ni en otros libros encontraremos detalles técnicos o esquemas eléctricos. Por otra parte, esta información sólo es necesaria para quien quiere modificar o completar el «hardware» de su ordenador, o para quien diseñe, por ejemplo, un interfaz para determinado periférico.
Salvo este punto, el usuario avanzado puede encontrar el resto de la información, fundamentalmente la descripción de las ROMs y de los resgistros de entrada-salida (E/S), seleccionándola de numerosos libros, ya que no existe uno exhautivo sobre este tema.
El libro que nos ocupa puede ser uno importante en esta selección, cubriendo casi por completo algunos temas, mientras que otros están tratados sucintamente.
Los dos primeros capítulo, de los ocho de que consta el libro, están dedicados a la programación en código máquina. Van dirigidos a los programadores que ya tienen algunos conocimientos de este lenguaje, por lo que no se hace más que una breve descripción de las instrucciones del 6510, y sin embargo se tratan otros temas, como la E/S desde programas en ensamblador, la programación del interface RS-232 y el bus serie. Se explica cómo trabajar con un monitor y un ensamblador típicos, aunque sin ofrecer el código de ninguno, y se comentan las direcciones más útiles de las ROMs del 64.
En el capítulo 3 se hace una breve descripción de la arquitectura del sistma, insistiendo particularmente en las difernetes configuraciones de memoria.
Los dos siguientes capítulo está dedicados respectivamente al sonido y a los gráficos. Ambos siguen la misma orientación: describen los registros e ilustran con ejemplos como deben utilizarse. Algunos de éstos son muy interesantes, y la mayoría están escritos en ensamblador. Es especialmente completa la descripción de los sprites en el capítulo 5, y el dedicado a sonido incluye una explicación de los convertidores A/D.
El capítulo 6 se centra en el intérprete BASIC. Comenta cómo trabaja, cuáles son sus principales rutinas, las direcciones más importantes de memoria, y como añadir nuevos comandos. Es uno de los capítulos más interesantes. La información que contiene permitirá al lector acceder a rutinas muy útiles del intérprete.
El siguiente capítulo compara el C-64 y el VIC-20. Aunque muy breve, puede ser interesante para los usuarios que quieren adapatar programas de uno a otro microordenador.
En el último capítulo se describen los chips CIA 6526, básicamente los «ports» y temporizadores que contienen, resolviendo alguno de los problemas que plantea la exactitud del reloj TI$.
Aproximadamente la mitad del libro está ocupada por el apéndice A, que contienen el listado ensamblador de las ROMs del 64, con una descripción más completa que la que se encuentra en otros libros.
A lo largo del texto se proponen algunas aplicaciones que algunos lectores encontrarán interesantes.
En definitiva un buen libro para los usuarios que desean conocer más a fondo su 64.

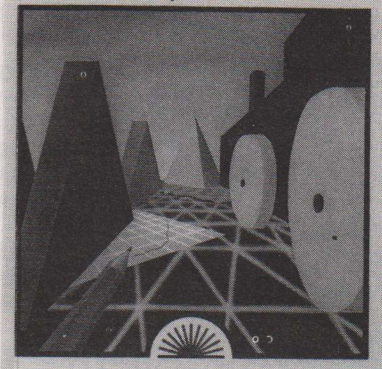
PROGRAMMING FOR EDUCATION ON THE COMMODORE 64

Autores: John Scriven y Patrick Hall
Editorial: Sunshine Books.
140 páginas.

El tema de la introducción del ordenador en las escuelas se ha puesto de moda en nuestro país. Son numerosos los planes privados y públicos que pretenden acercar el instrumento informático a profesores y alumnos. Sin embargo, la producción de «software» y libros para este sector es todavía muy reducida.

El C-64 es uno de los microordenadores que cuenta con más «software» educativo, siempre escaso, pero no puede decirse lo mismo de la bibliografía al respecto. Este libro tiene por tanto el interés añadido de ser una novedad.
Está dirigido a profesores de la primera etapa de enseñanza básica, con un doble objetivo: introducir al profesorado en la microinformática y aportar ideas sobre cómo ordenador puede contribuir a la educación de los niños.

Este doble objetivo se intenta cumplir a través de una colección de programas didácticos, breves, cuyo interés radica en la idea de cada programa y el estilo



modular de programación empleado
Los programas podríamos dividirlos, según la facultad que pretenden desarrollar en el niño, en los siguientes grupos:

— De desarrollo de la habilidad aritmética.

— De ampliación del vocabulario y dominio del idioma.

— De desarrollo del razonamiento lógico.

— De juegos.

Estos son algunos de los mejores programas del libro:

— «Fraction»: Programa para hacer comprender al niño el significado de una fracción mediante una representación gráfica de la misma.

— «Alphasort»: El alumno debe ordenar alfabéticamente un conjunto de palabras en un tiempo récord.

— «Simon»: Este conocido juego pretende desarrollar la memoria del niño, que debe repetir en el mismo orden una secuencia, cada vez más larga, de números elegidos aleatoriamente por el ordenador.

— «Anagram»: Se trata de completar una palabra escogida por el ordenador, a partir de algunas de sus letras.

— «Story»: El ordenador elige un tema y pregunta al alumno adjetivos asociados y palabras preseleccionadas, que pueden resultar divertidas. Se trata de ampliar el vocabulario mientras se juega.

— «Treasure»: Para localizar un tesoro que el ordenador a escondido en la pantalla, el niño debe seguir una secuencia de operaciones. Sólo si emplea una estrategia lógica podrá alcanzar el objetivo.

En general, todos los programas premian al niño con un dibujo o una pequeña melodía si ha tenido éxito o invitan a hacerlo mejor en caso contrario.

Todos los programas están escritos en un BASIC sencillo y explicados de manera asequible para los que se inician en este lenguaje.

José M.ª Lizaso Azcárate



Una ayuda con las datas

Esta corta rutina puede ser una ayuda inestimable a la hora de trabajar con sentencias DATA. Lo que hace es sencillamente transformar cualquier zona de memoria, que se especificará mediante sus direcciones de comienzo y de final, en un conjunto de sentencias DATA cuyos valores son los contenidos de la zona de memoria. Por ejemplo, supongamos que hemos estado diseñando un sprite y lo tenemos almacenado a partir de la posición 24000 de memoria. Podremos convertir la información sobre dicho sprite en un conjunto de sentencias DATA instantáneamente sin más que utilizar la rutina. Otro caso en el que nos puede resultar muy útil es a la hora de incorporar rutinas en lenguaje máquina a nuestros programas BASIC. Sin más que saber la dirección de comienzo de la rutina y su longitud, podremos incorporarla a nuestro programa en cuestión de segundos. El funcionamiento de la rutina es

```
1 POKE53280,6:POKE53281,6
2 CLR:PRINTCHR$(147)
3 INPUT"PRIMER NUMERO DE LINEA";LN
4 IFLN<30 OR LN>63983THEN2
5 INPUT"DIRECCION DE INICIO";S
6 INPUT"DIRECCION FINAL";F
7 IFS>=FTHENPRINT:PRINT"ERROR EN LA DIRECCION":GOTO 4
8 IFF-S>144THENPRINT"MAX 144 BYTES":GOTO4
9 PRINTCHR$(147)CHR$(17)
10 PRINTLN;"DATA";:C=C+1
11 FORY=STOS+16
12 IFY>FTHENPRINTCHR$(157)" ":GOTO20
13 P=PEEK(Y):P$=STR$(P)
14 P$=RIGHT$(P$,(LEN(P$)-1))
15 IFY<>S+16THENP$=P$+","
16 PRINTP$;:NEXTY:PRINT
17 S=S+17:LN=LN+2
18 IFS>F OR C=9THENGOTO20
19 GOTO 10
20 PRINT"GOTO 24"
21 FORY=1TOC+1
22 POKE630+Y,13:NEXTY
23 POKE198,C+1:PRINTCHR$(19);:END
24 PRINT:PRINT"INTRODUCIDAS LINEAS DATAS"
```

como sigue. Después de teclearla y al escribir RUN se nos preguntará por «el primer número de línea». Este número será el correspondiente a la primera línea de DATAS y queremos incorporarle una serie de sentencias DATA que comiencen en la línea 2000 contestaremos con dicho número a la primera pregunta. A continuación habrá que introducir las direcciones de comienzao y de terminación de la zona de memoria a convertir. Estas direcciones se introducirán en decimal.

Tras esto la rutina entrará en funcionamiento y transformará la zona de memoria en sentencias DATA que incluirá en nuestro programa. Ahora lo único que queda es borrar las líneas que corresponden a la rutina. El máximo número de valores que pueden transformarse en una pasada es de 144, pero se pueden llevar a cabo tantas pasadas como se quiera sin más que tener en cuenta que en cada una habrá que modificar el primer número de línea.

Alta resolución y velocidad

Todo el que haya trabajado con la alta resolución del C-64 desde BASIC sabe que el proceso es en general muy lento. Ello se debe a que antes de poder empezar con el dibujo de las líneas, puntos o lo oque se quiera dibujar hay que encargarse del borrado de la pantalla de alta resolución, que normalmente se encuentra llena de lo que se llama «basura». Para ello hay que llenar de ceros una zona de memoria de 8000 bytes. Desde BASIC esto significa 8000 POKES y ello consume bastante tiempo. Además hay que encargarse del color. La información de color de la alta resolución se encuentra en la zona de memoria que corresponde a la pantalla de baja resolución y que comienza en la dirección 1024. En este caso, para corregir adecuadamente esta información hay que modificar los valores de nada menos que 1000 posiciones de memoria. Más tiempo que se consume. Por ello siempre resulta interesante cualquier

```
10 FORI=0TO98:READA:POKE49152+I,A
15 T=T+A:NEXT:IFT<>14402THENPRINT"ERROR EN DATAS":END
20 HI=49152:CL=49180:CO=49210:SC=8192
30 SYSHI:SYSCL:POKE255,16:SYSCO
40 END
100 DATA169,32,133,52,133,56,169,147
110 DATA32,210,255,173,24,208,9,8,141
120 DATA24,208,173,17,208,9,32,141,17
130 DATA208,96,169,0,133,251,169,32
140 DATA133,252,166,252,160,0,169,0
150 DATA145,251,200,208,251,232,224,64
160 DATA240,5,134,252,76,36,192,96,169
170 DATA0,133,253,169,4,133,254,165
180 DATA255,166,254,160,0,145,253,200
190 DATA208,251,232,224,7,240,5,134
200 DATA254,76,70,192,160,0,230,254
210 DATA145,253,200,192,233,208,249,96
```

rutina que utilizando la velocidad de los programas en lenguaje máquina se encargue de este penoso trabajo. Esta es la función de la rutina que os ofrecemos. Su forma es, como siempre, la de un cargador BASIC que se encarga de cargar el código máquina en memoria.

Una vez cargada en memoria la rutina máquina en el programa la llama mediante varios SYS que son los de la línea 30. Si se incorpora la rutina como parte de otro programa BASIC se podrán utilizar estos mismo SYS. El borrado de la pantalla y el limpiado de la zona de memoria del color son casi instantáneos, el ahorro de tiempo es impresionante. Un último deta-

lle lo constituye el POKE que aparece en la línea 30 antes del SYS CO. Este POKE es el que determina los colores del fondo y de la pantalla de alta resolución y es necesario incluirlo siempre antes del SYS.

Realimentación del teclado en el C-64.

Bajo este título en principio un tanto complejo se esconde una idea muy simple. En determinados ordenadores, existe la posibilidad de asociar a cada tecla o grupo de teclas un tono o un sonido, de forma que al pulsar dicha tecla el tono se escuche. De estee modo, al pulsar cualquier tecla se recibe información sobre si la tecla ha sido realmente pulsada. La rutina que os ofrecemos proporciona tres tonos diferentes. El tonò más agudo corresponde a la tecla de RETURN, un tono medio indica que se han pulsado las teclas de cursor o alguna de las teclas de función y, por último, un tono grave es lo que se puede escuchar al pulsar cualquiera de las teclas alfanuméricas.

La rutina es un clásico cargador BASIC que introduce el programa en lenguaje máquina directamente en la memoria del ordenador.

Después de escribir RUN se podrán escuchar los diversos tonos cuando se pulse alguna de las teclas. La rutina se carga a partir de la posición 50000 de la memoria, por lo que para activarla no hay más que teclear SYS 50000. Desactivarla resulta más fácil todavía ya que puede hacerse con la secuencia RUN/STOP RESTORE. Al ejecutar la rutina por primera vez ella sóla se activa, desde el cargador BASIC, a través de la línea 80 que incluye el citado SYS. La rutina podrá utilizarse independientemente o como parte de otro programa, por ejemplo en las fases introducción de datos.

```
10 REM *** REALIMENTACION DEL TECLADO ***
20 POKE 54275,8 :REM AMPLITUD DE ONDA
30 POKE 54277,36:REM ATAQUE/DECAIMIENTO
40 POKE 54278,9 :REM SOSTENIMIENTO/RELAJACION
50 POKE 254,65  :REM FORMA DE ONDA
51 :
60 FOR D=0 TO 54 :READ MCODE :POKE 50000+D,MCODE:T=T+MCODE:NEXT
70 IF T<>6539 THEN PRINT"REVISA DATAS":END
80 SYS 50000
90 DATA 120,169,98,141,20,3,169,195,141,21,3,88,169,15,141,24,212,96,165
100 DATA 197,164,254,136,201,64,240,22,162,128,201,1,240,12,162,64,201
110 DATA 8,144,6,201,51,240,2,162,16,142,1,212,200,140,4,212,76,49,234
```



Rotundamente no

P: Miguel Angel Alvarez (Segovia).

Ante la publicación por Vds., en el n.º 19 de su revista, del artículo titulado "pon un reset en tu Commodore", y considerándolo interesante, procedí a su instalación en mi ordenador. Comprobando que su montaje y funcionamiento eran correctos, hecho que constate durante unos tres días, transcurridos los cuales surgió el siguiente fallo en el ordenador: dejaron de funcional el grupo de letras 3-4, W-E,A-S y Shift-A. Circunstancia que atribuí a un posible fallo del ordenador, el cual solo hace dos meses que le compré.

Pero mi sorpresa y duda ha surgido cuando a un amigo mío, que también ha efectuado el montaje de esta utilidad, le han dejado de funcionar también a los varios días, el movimiento de subida y bajada en el port n.º 2 solamente, y en juegos con los que con anterioridad había jugado con asiduidad.

Por estos hechos paso a plantearles las siguientes preguntas:

1.- ¿Creen Vds. que estos fallos son de los ordenadores?

2.- ¿Es posible que se hayan producido por la instalación del reset?

3.- ¿Los montajes que Vds. publican están suficiente probados y tienen total garantía de su buen funcionamiento?

R: El NO rotundo con el que titulamos la respuesta a esta carta se refiere a la posibilidad de que el montaje del pulsador de RESET haya podido causar daños en cualquier ordenador.

El montaje, siempre que se haya realizado siguiendo las indicaciones del artículo, es totalmente inocuo e incapaz de causar ningún daño. De hecho es el tipo de montaje que incorporan de fábrica numerosos ordenadores. Esto responde a la segunda de las preguntas, en cuanto a las otras dos hay que decir lo siguiente: El montaje, antes de su publicación, fue exaustivamente probado, comprobándose su perfecto funcionamiento. Actualmente el montaje sigue siendo funcionando a la perfección sin que haya habido el más mínimo problema con ninguna de las teclas del teclado. Por otra parte, es díficil establecer un diagnóstico sobre si el fallo es del teclado o puede ser debido a problemas internos en alguno de los chips. Lo más aconsejable es acudir a un buen servicio técnico en el que, con los medios adecuados, podrán encontrar la avería.

CP/M Cobol...

P: Josep Sanahuja (Barcelona)

Soy suscriptor de su revista y poseedor de un Ordenador Personal "COMMODORE", amen de diferentes periféricos y accesorios.

Me dirijo a Vds. esperando puedan solucionarme las siguientes preguntas:

1.- Puede mi ordenador adaptaséle algún cartucho a fin de poder operar con el Sistema Operativa CPM o CMP/86.

2.- Puede igualmente con otro cartucho ejecutar y compilar programas en COBOL.

3.- De ser así, donde se pueden encontrar los citados cartuchos.

R: Existe una versión del sistema operativo CP/M, de la propia casa Commodore, que se lanzó hace ya bastante tiempo y que incluía un cartucho a conectar en el port de expansión y un diskette con el sistema operativo. En el interior del cartucho se encontraba el microprocesador Z-80, necesario para el funcionamiento del conjunto. Sin embargo, el número de programas CP/M adaptados al formato de los diskettes de Commodore es, que nosotros sepamos, muy escaso. Para más información sobre este cartucho CP/M lo más conveniente es dirigirse a Microelectrónica y Control, distribuidores del mismo.

En cuanto al CP/M 86 y al lenguaje COBOL, no tenemos noticias de que exista ninguna versión para las máquinas de Commodore, excepto para los ordenadores de la gama profesional PC 10 y Pc 20.

Tres preguntas sobre el 64

P: Ramón Altisent (Tarragona)

Poseo un Commodore 64 y hago con él mis propios programas. Solo hace un año y medio que lo tengo y he aprendido bastante, pero hay tres cosas que no tengo claras y me gustaría que me respondierais:

1.- ¿Como se hacen los gráficos en BASIC.

2.- ¿Que es lo que hay que hacer para que tus programas no se puedan listar ni copiar?

3.- ¿Es posible hacer hablar a un Commodore?

Por favor, os pido que me contestéis a través de la revista, cuanto antes mejor, gracias.

R: Vamos a darte unas respuestas de carácter general que esperamos sirvan para aclarar conceptos tan generales como los que nos planteas. En primer lugar, el proceso de creacción de gráficos desde lenguaje BASIC es, desgraciadamente, una tarea ingrata en el Commodore 64 ya que no proporciona comandos específicos para el trabajo con gráficos y hay que recurrir a la utilización de múltiples PEEKs y POKEs a memoria. En cualquier caso siempre habrá que definir que es lo que se quiere hacer y elegir en consecuencia entre los diversos modos gráficos. Por ejemplo habrá que definir si se desea trabajar en alta o baja resolución, si los gráficos van a ser estáticos o se van a mover (en este último caso lo mejor es recurrir a los sprites) y en fin, toda una serie de consideraciones que vendrán definidas por el tipo de gráficos que se deseen realizar: En todos los casos, una vez que se haya definido el problema habrá que recurrir o a programas gráficos especializados o a un buen manual.

En cuanto a la protección de los programas hay que decir que se trata de un tema complejo en el que cabe hablar de multitud de métodos de protección, unos más sencillos, otros más complicados y, eso sí, ninguno de ellos infalible. Por ejemplo, uno de los métodos más simples de protección, ya comentado en nuestras páginas, consiste en inhibir al comienzo del programa la tecla RUN/STOP lo que imposibilita parar el programa. Por último, claro que es posible hacer hablar a un Commodore. Para ello no hay más que recurrir a alguno de los múltiples sistemas de síntesis de voz que se pueden encontrar en el mercado, algunos de los cuales comentamos en el número 15 de Commodore Magazine.



Desearía intercambiar programas e información para el C-64, únicamente en disco, con usuarios que dispongan de una programoteca superior a 2.000 títulos. Dirigirse a: Francisco J. Lo Perena. Daoiz, 2. Ceuta. Telf. (956) 51 57 19.

Vendo ordenador Commodore CBM-4032 junto con unidad de discos doble CBM-8050, todo ello en perfecto estado. Además, con un stock de más de 150 programas (utilidades, juegos, bases de datos), así como todos los cables para su interconexión. Precio total 280.000 ptas. (precio real 450.000 ptas.). Interesados dirigirse a: José Bermell Tarazona. Plus Ultra, 23, 3.º 46006 Valencia. Telf. (96) 334 03 40. A partir de las 7.

Vendo los programas siguientes o cambio por programas para el Commodore 64: Contabilidad y Facturación para Commodore de la serie 3000 con completas instrucciones. Contabilidad para CBM de las series 4000 y 8000. Base de datos para CBM series 4000 y 8000. Todo a precios de ocasión. Dirigirse a: Fernando Pérez Rodrigo. C/ Venezuela, 29, 3.º B. 46007 Valencia. Telf. (96) 377 25 61. Tardes.

Soy un usuario del Commodore 64 que está interesado en contactar con amigos de toda España para intercambio de todo tipo de programas. Los interesados enviar lista por correo o contactar directamente conmigo. Carlos Bermell Molmeneu. C/ Plus Ultra, 23. 46006 Valencia. Telf. (96) 377 25 61. Tardes.

Me llamo Javier y estoy interesado en adquirir una unidad de disco, que funcione bien, por un lote de programas de 300 títulos, de primera calidad. También haría intercambios. Interesados dirigirse a: Javier Hernández Sole. C/ El Roser, 34. Mora de la Nova (Tarragona). Telf. (977) 40 10 84.

Desearía intercambiar cualquier programa en cinta para Commodore 64. Tengo juegos de alta calidad en gráficos y acción, también tengo programas de utilidades que son bastante buenos para la ayuda de programación y para poder sacar mucho provecho al ordenador. Interesados escribir a: Miguel Angel Rico Duro. C/ Mallorca, 28-B, 1º, 1ª. 43005 Tarragona.

Si quieres tener los mejores y últimos programas para tu Commodore 64 entra en Intersoft. El club de intercambio de *software* para el C-64. Intercambiamos programas, trucos, ideas, experiencias, etc... Solicita información a Manuel Arroyo. 27016 Barcelona.

Estoy interesadísimo en intercambiar juegos para C-64 en *cassette*. Tengo unos 150 juegos buenísimos. Si quieres intercambiar conmigo, rápidamente escribe a: Juan Alberto Cort. Rambla Nova, 77, 1º, 1ª. 43003 Tarragona.

Club (C) LDA. Intercambio de Experiencias y *software*, a clubs y particulares. Constestación garantizada. Escribir a: Andreu Ibañez i Perales. C/ Passeig de Ronda 76, 3º, 1ª. 25006 Lleida.

Vendo ordenador VIC-20 en perfecto estado, un curso de introducción al BASIC con sus cintas correspondientes y un cartucho de ajedrez Sargón II; todo por 20.000 ptas. Interesados dirigirse a José Gómez, C/ Bobiles, 70. Hospitalet del Llobregat. (Barcelona). Tel.: 334 90 98.

Deseo intercambiar programas para C-64 en cinta o disco. Poseo más de 500. Mandar lista a: Jose A. Guilabert. Urbanización Puerto Sol. C/ Barquero, 27. Roquetas de Mar. (Almería). Prometo contestar.

Desearía cambiar toda clase de programas y experiencias para el CBM-64 en disco. Prometo contestar. Dirigirse a: Carlos de la Presa Gómez. Montepinar n.º 8. Guadarrama (Madrid).

Cambio 15 juegos y 5 utilidades por uno de estos programas: Oxford Pascal, Zoom Pascal, Summer Games, Entombed o Staff of Karnack, para Commodore 64. Los interesados enviad cinta con uno de estos programas a: Miguel Angel Rico Duro. C/ Mallorca, 28-B, 1º, 1ª. 43005 Tarragona. Recibirán de inmediato los 15 programas y 5 utilidades en cinta.

Vendo los programas en disco: BASIC LIGHTNING, aporta 150 comandos (5.000 ptas.); WHITE LIGHTNING, lenguaje FORTH con 300 comandos (5.000 ptas.) ambos con instrucciones. SIMON'S BASIC, aporta 115 comandos; VIC-TREE, aportas 42 comandos (5.000 ptas cada uno). BLITZ COMPILER, compila programas BASIC a Código Máquina (2.500 ptas). 80-COLUMNAS (2.500 ptas). Todos ellos en disco. Cambiaría toda clase de SOFT para CBM-64 en disco o *cassette*, tanto juegos como utilidades. Llamar de 1 a 3 al tel.: (91) 718 36 17. Preguntar por Francisco Javier.

VIC-20, con ampliación de 16K, tarjeta de carga rápida VIC-SPEED, cartucho "SLOT", libros: "Guía del usuario" (2 tomos). Llamar a Rafael Rojo al tel.: 254 80 62.

Cambio: 201 juegos (Misión Imposible, Staff of Karnack, Match Point...) y 34 programas de aplicaciones (Logo, Pascal, Simon's...) para C-64 por unidad de disco o impresora. También cambio programas. Interesados dirigirse a: Fernándo Lozano Bojadós. C./ Renclusa, 51, 6º, 1ª. Hospitalet del Llobregat. (Barcelona). Tel.: 249 43 96.

Vendo C-16 por razones personales junto con su *datassette* y dos juegos. Todo por 49.000 ptas. Hace seis meses que lo compré y lo garantizo tres meses. Interesados dirigirse a: Alfonso Janeiro Díez. C/ Vial III Boa Vista. Chalet 14. Pontevedra.

Cambio juegos y utilidades para C-64 en disco. Tengo más de 300. Dirigirse a: Enric Corominas Bosh. C/ Ultónia, 8, 6º 2ª. 17002 Girona.

Vendo C-64, unidad de discos 1541 y 20.000 ptas. en programas de todo tipo. Todo en perfecto estado, por 150.000 ptas. Interesados preguntar por David en el tel.: 734 11 03. Madrid.

Vendo Vic-20, cartucho Superexpander + 3K, cartucho de ayuda al programador, ampliación de memoria 32K Ram, *cassette* original Commodore C2N. Interesados dirigirse a: César Toquero López. C/ Santucho, 73, 4ºC. 48006 Bilbao. (Vizcaya).